Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Luka Zakrajšek

Vizualizacija in analiza glasbenih posnetkov s kompozicionalnim hierarhičnim modelom

diplomsko delo

univerzitetni študijski program prve stopnje
računalništvo in informatika

Mentor: prof. dr. Saša Divjak
Somentor: doc. dr. Matija Marolt

Ljubljana 2015

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V diplomski nalogi izdelajte orodje za vizualizacijo in analizo glasbenih posnetkov z uporabo kompozicionalnega hierarhičnega modela. Model preizkusite na primerni zbirki skladb in poskusite izboljšati njegovo natančnost.

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Luka Zakrajšek sem avtor diplomskega dela z naslovom:

Vizualizacija in analiza glasbenih posnetkov s kompozicionalnim hierarhičnim modelom.

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Saše Divjaka in somentorstvom doc. dr. Matije Marolta,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 21. januarja 2016

Podpis avtorja:

Zahvaljujem se obema mentorjema za vse nasvete in pomoč. Posebna zahvala gre tudi asistentu Matevžu Pesku za ideje, usmerjanje, spodbude, razlage in moralno podporo. Zahvaljujem se tudi kolegom, prijateljem in svoji družini.

Kazalo

Povzetek

Abstract

1 Uvod **1**

2 Pregled področja **3**

3 Kompozicionalni hierarhični model **5**

3.1 Relativnost in deljivost delov 8

3.2 Analiza 9

3.3 Učenje 14

4 Nenegativna matrična faktorizacija **17**

4.1 Delovanje NMF 18

4.2 NMF za ocenjevanje več tonskih višin 20

4.3 Diskriminativni kriterij 21

4.4 Avtokodirni model 23

5 Vizualizacija **27**

5.1 Predstavitev glasbe 27

5.2 Spletna storitev 28

5.3 Spletna aplikacija 30

6 Izboljšava rezultatov **41**

6.1 Odstranjevanje osamelih dogodkov 41

6.2 Glajenje prekinjenih dogodkov 41

6.3 Odstranjevanje višjih harmonikov 42

6.4 Odstranjevanje dogodkov izven vokalnega razpona 42

7 Evalvacija **43**

7.1 Ocenjevanje klasifikatorjev 43

7.2 Zbirke 46

7.3 Testiranje 46

7.4 Analiza in diskusija 57

8 Zaključek **59**

8.1 Prednosti in omejitve orodja 59

8.2 Nadaljnje delo 60

Literatura **61**

Seznam uporabljenih kratic

kratica	angleško	slovensko
AGC	Automatic gain control	samodejno uravnavanje jakosti
CHM	Compositional Hierarchical Model	kompozicionalni hierarhični model
DNMF	Discriminative Non-negative Matrix Factorization	diskriminativna nenegativna matrična faktorizacija
HTTP	HyperText Transfer Protocol	protokol za prenos hiperteksta
JSON	JavaScript Object Notation	objektna notacija za JavaScript
MAPS	MIDI Aligned Piano Sounds	poravnani klavirski zvoki MIDI
MIDI	Musical Instrument Digital Interface	glasbeni instrumentalni digitalni vmesnik
MIR	Music Information Retrieval	pridobivanje informacij iz glasbe
MVVM	Model-View-View-Model	model - pogled - model za pogled
NMF	Non-negative Matrix Factorization	nenegativna matrična faktorizacija
PCM	Pulse Code Modulation	pulzno kodna modulacija
REST	Representational State Transfer	Prenos predstavljajočega stanja
SVM	Support Vector Machine	metoda podpornih vektorjev
WAV	WAVeform audio format	zvočni format, ki uporablja valovne krivulje
XML	eXtensible Markup Language	razširljiv označevalni jezik

Povzetek

V diplomski nalogi razvijemo orodje za vizualizacijo in analizo glasbenih posnetkov z uporabo kompozicionalnega hierarhičnega modela. Model omogoča učenje konceptov tonov iz monofoničnih posnetkov, transparenten vpogled v naučene strukture ter robustno in hitro obdelavo zvočnih posnetkov. Nadgradimo ga z metodo diskriminativne nenegativne matrične faktorizacije. S to metodo se lahko zelo dobro prilagodimo polifoničnim posnetkom. Uvedli smo tudi različne tehnike čiščenja hipotez o prisotnosti tonskih višin, s katerimi izboljšamo končne rezultate. Model preizkusimo na zbirki polifoničnih klavirskih posnetkov, vokalni zbirki ljudskih pesmi ter na sintetizirani zbirki različnih inštrumentov. Z modelom CHM in nadgradnjo DNMF dobimo zelo dobre rezultate, zato model uporabimo kot osnovo za spletno aplikacijo. Aplikacija omogoča nalaganje novih zvočnih posnetkov, učenje in testiranje novih modelov, grafično predstavitev naučenih struktur ter pogled klavirske tabulature. Slednji omogoča analizo pridobljenih transkripcij, interaktivno urejanje in dodajanje ritmičnih anotacij ter izvoz v druga orodja.

Ključne besede: kompozicionalni hierarhični model, nenegativna matrična faktorizacija, pridobivanje informacij iz glasbe, transkripcija.

Abstract

This thesis provides a tool for visualization and analysis of music recordings using compositional hierarchical model. Model learns the concept of music tones from monophonic recordings, transparent insight into learned structures and also robust and fast processing of sound recordings. Model is extended with discriminative non-negative matrix factorization method. With this method we can get a really good fit for polyphonic recordings. We introduced various techniques for pitch hipothesis cleaning that improve final results. Model is evaluated on polyphonic piano recording database, vocal collection of folk music and synthesized collection of various instruments. We achieve very significant results using CHM and DNMF and use CHM as a basis for the web application. Application can be used to upload new sound recordings, learn and test new models, observe graphical representation of learned structures and piano roll view. Piano roll helps us analyze generated transcriptions, interactive editing, adding rhythmic annotations and export data for further manipulation using other software products.

Keywords: compositional hierarchical model, non-negative matrix factorization, music information retrieval, transcription.

Poglavje 1

Uvod

Glasbena transkripcija je postopek pretvarjanja zvočnega posnetka v notni zapis. Mnogim glasbenikom predstavlja izziv, saj se morajo osredotočiti na glasbo in ob poslušanju posnetkov zapisovati note, ki jih slišijo. Postopek je dolgotrajen, rezultati pa so uporabni za natančno ponovitev izvedbe glasbe, kot jo izvaja glasbenik na posnetku. Kakovost in hitrost transkripcije je omejena z glasbenikovim posluhom in znanjem. Včasih je zaradi prisotnosti velike polifonije ali nizkih leg težko ugotoviti, kaj sestavlja harmonijo. Za razpoznavanje prisotnih akordov je velikokrat dovolj že dober posluh, če pa se lotimo podrobne transkripcije kompleksne glasbe, lahko postopek za zapis ene minute posnetka traja več ur. Tukaj nam na pomoč priskočijo orodja, ki nam z analizo glasbenega spektra pokažejo, katere frekvence oz. tonske višine so prisotne v nekem časovnem odseku. Primer takega orodja je aplikacija Transcribe![1], ki omogoča pogled klavirske tabulature, vendar ne omogoča izvoza podatkov. Obstajajo tudi orodja kot na primer SONIC[2], ki nam glasbene posnetke samodejno pretvorijo v notni zapis oz. obliko MIDI, vendar pri teh orodjih velikokrat nimamo nadzora nad postopkom, napake pa moramo ročno popravljati v drugih orodjih.

Cilj diplomskega dela je razviti postopke in orodje za transkripcijo in

[1] `http://www.seventhstring.com/xscribe/overview.html`
[2] `http://lgm.fri.uni-lj.si/matic/SONIC/`

analizo zvočnih posnetkov. Za analizo zvočnih posnetkov in prisotnih tonskih višin bomo uporabili kompozicionalni hierarhični model za pridobivanje informacij iz glasbe [20]. Model je globoka arhitektura, ki omogoča transparenten vpogled v različne stopnje pridobivanja podatkov. Natančnost modela bomo za potrebe transkripcije izboljšali z uporabo diskriminativne nenegativne matrične faktorizacije [3] in rezultate primerjali z drugimi pristopi glede na natančnost in robustnost. Na osnovi kompozicionalnega modela bomo razvili orodje, ki nam bo pomagalo pri izdelavi transkripcij in analizi zvočnih posnetkov. Orodje bo omogočalo vizualni pregled in urejanje transkripcije z možnostjo izvoza pridobljenih informacij v standardno obliko MIDI.

Poglavje 2

Pregled področja

Prvi pristopi v pridobivanju informacij iz glasbe s pomočjo računalnikov segajo v leto 1962. Takrat je Bernard Gold [8] v znanstveni reviji Journal of the Acoustical Society of America objavil članek z naslovom *Computer Program for Pitch Extraction*, v katerem je predstavil računalniški program za odkrivanje tonskih višin. Sondhi [24] je leta 1968 v članku *New Methods of Pitch Extractions* predstavil tri nove metode za pridobivanje osnovnih frekvenc: sploščevanje spektra z minimalnimi popravki faze za uskladitev harmonikov, sploščevanje spektra s samodejno korelacijo in nelinearno deformacijo s samodejno korelacijo. Leta 1977 je Moorer [18] opisal pridobivanje tonskih višin in samodejno zapisovanje not z uporabo pasovnoprepustnih filtrov. Pokazal je dva primera, ki prikazujeta uspešnost sistema glede na omejitve analizirane glasbe.

V zadnjih letih se na področju veliko uporablja metoda nenegativne matrične faktorizacije [13] (NMF). Že leta 2004 jo Abdallah [1] uporabi za polifonično transkripcijo, kasneje pa tudi Cont [5], Raczyński [22] in drugi. Klapuri [12] predstavi ocenjevanje osnovnih frekvenc s ponavljanjem iskanja tonov in odstranjevanja pripadajočih frekvenc iz mešanice. Poliner in Ellis [21] leta 2006 predstavita diskriminativni model za polifonično klavirsko transkripcijo z uporabo metode podpornih vektorjev (SVM). V modelu so izhodi klasifikatorjev začasno omejeni z uporabo skritih Markovih modelov, sistem

pa je uporaben za transkripcijo posnetkov pravih klavirjev in sintetizatorjev. Boulanger [3] leta 2012 predstavi metodo diskriminativne nenegativne matrične faktorizacije (DNMF), nadgrajeno s SVM. Metodo DNMF bomo uporabljali tudi v nadaljevanju. Leta 2013 Weninger [26] predstavi nov pristop z uporabo SVM in NMF, kjer posamezne klavirske tone časovno razdeli na več delov.

Zadnji prispevki na področju uporabljajo strojno učenje z uporabo nevronskih mrež in ostalih globokih arhitektur. Marolt [17] leta 2004 predstavi novo tehniko sledenja delov, ki temelji na kombinaciji avditornega modela in prilagodljivih oscilatornih mrež. Predstavi tudi orodje za izdelavo transkripcij SONIC [16], ki je prostodostopno in na voljo za primerjavo rezultatov. Pesek [20] predstavi kompozicionalni hierarhični model, ki s svojo robustnostjo poleg klavirske glasbe omogoča tudi transkripcijo drugih inštrumentov in vokalne glasbe ter ocenjevanje akordov.

Obstajajo obsežne zvočne zbirke (npr. zbirka MAPS [6, 7] in zbirka slovenskih ljudskih pesmi[1]), posnete na studijskih klavirjih in sintetizatorjih, ki vsebujejo anotacije, z uporabo katerih lahko izračunamo točnost pridobljenih transkripcij.

[1] `http://www.etnofletno.si/`

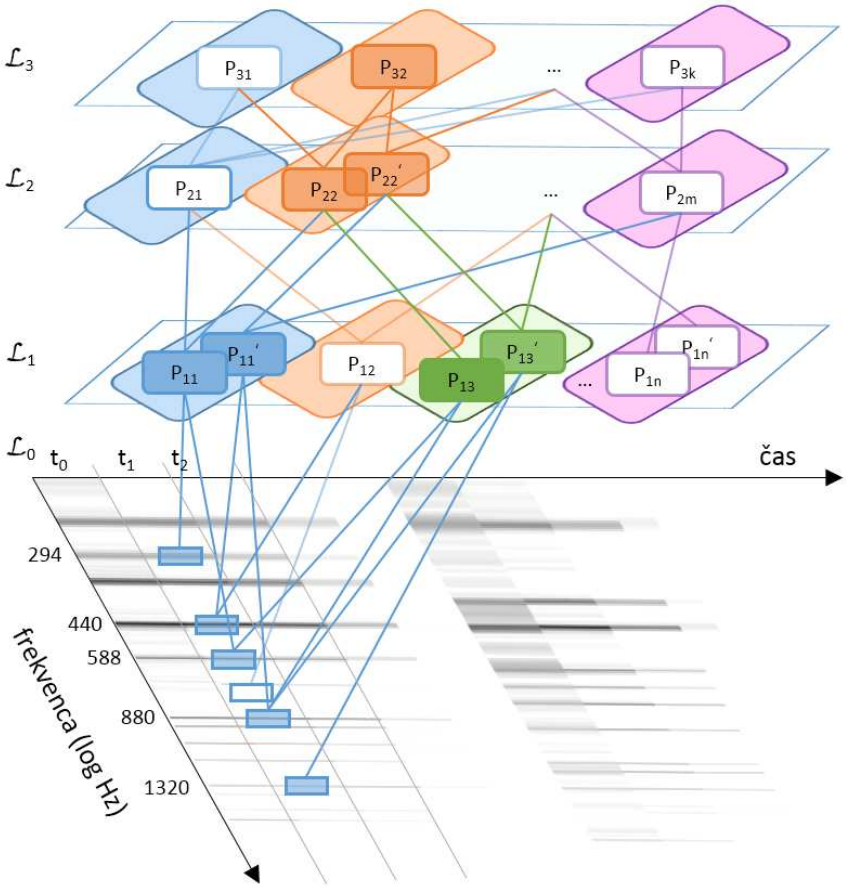
Poglavje 3

Kompozicionalni hierarhični model

Kompozicionalni hierarhični model je globoka arhitektura (ang. deep architecture) s transparentno strukturo, ki omogoča predstavitev in razlago vsebine signalov na različnih nivojih kompleksnosti. Model so v Laboratoriju za računalniško grafiko in multimedije na Fakulteti za računalništvo in informatiko razvili Pesek et. al [20]. V nadaljevanju povzemamo definicijo modela, predstavljenega v članku [20].

Model ima zmožnost nenadzorovanega učenja hierarhične predstavitve zvočnega signala od komponent signala na najnižjem nivoju do posameznih glasbenih dogodkov na najvišjih nivojih. Zgrajen je na predpostavki, da se lahko kompleksni signali razstavijo v hierarhijo gradnikov - delov. Ti deli se pojavljajo na različnih stopnjah razdrobljenosti in predstavljajo množice entitet, ki opisujejo signal. Glede na svojo kompleksnost se lahko deli pojavljajo na več nivojih, od manj do bolj kompleksnih. Deli na višjih nivojih so izraženi kot kompozicije delov na nižjih nivojih (npr. akord je sestavljen iz več tonov, vsak ton iz več harmonikov itd.). Del lahko torej opisuje posamezne frekvence v signalu, njihove kombinacije, posamezne tone, akorde in časovne vzorce, kot so zaporedja akordov.

Kompozicionalni hierarhični model je sestavljen iz več nivojev. Vsak nivo



Slika 3.1: Kompozicionalni hierarhični model. Deli na vhodnem nivoju ustrezajo komponentam signala v časovno-frekvenčni predstavitvi. Deli na zgornjih nivojih so kompozicije nižjenivojskih delov (označeni kot povezave). Del je lahko vsebovan v več kompozicijah, npr. P_{11} na prvem nivoju je del kompozicij P_{21}, P_{22} in P_{2m} na drugem nivoju. Več upodobitev istega dela (npr. P'_{11}) označuje več aktivacij dela na različnih lokacijah (vsi primeri dela na nivoju so označeni z isto barvo). Deli, aktivirani v času t_1, so prikazani s posameznimi barvami. Vir [20]

vsebuje množico delov. Posamezni del je kompozicija dveh ali več drugih delov z nižjega nivoja in je lahko vsebovan v poljubnem številu kompozicij na višjem nivoju. Kompozicionalni model torej tvori hierarhijo delov, kjer

vsak del predstavlja kompozicijo delov s spodnjih nivojev, kot je prikazano na sliki 3.1.

1. Vhodni nivo: vhodni nivo modela L_0 izvira iz časovno-frekvenčne predstavitve glasbenega signala. Vsebuje posamezne atomične dele, ki se aktivirajo (ustvarijo izhod) na lokacijah vseh frekvenčnih komponent v signalu v danem času. Primer je prikazan na sliki 3.1 (zaradi jasnosti so prikazane samo nekatere aktivacije v t_1). Bolj formalno je aktivacija dela definirana z dvema vrednostima: lokacijo L_P, ki ustreza frekvenci in magnitudo A_P, ki ustreza magnitudi frekvenčne komponente. Vhodni nivo je lahko kakršnakoli časovno-frekvenčna predstavitev, vendar logaritmični frekvenčni razmiki ustvarijo bolj kompaktne modele zaradi relativne narave kompozicije delov na višjih nivojih.

2. Višji nivoji: višji nivoji modela L_n vsebujejo množice kompozicij - delov, sestavljenih iz delov z nižjih nivojev. Vsaka kompozicija lahko vsebuje poljubno število delov s spodnjih nivojev (zaradi jasnosti razlage modela so uporabljene samo kompozicije z dvema deloma). Kompozicija je lahko vsebovana v poljubnem številu kompozicij na zgornjih nivojih. Na sliki 3.1 so označene kot povezave.

Kompozicija i na nivoju L_n je formalno definirana kot struktura, ki vsebuje dele z nivoja $(n-1)$: centralni del C in sekundarni del S. Deli, ki sestavljajo poddele kompozicije, so poimenovani. Kompozicija je definirana kot:

$$P_{n,i} = \{C_{n-1,j}, S_{n-1,k}, (\mu_{n,i}, \sigma_{n,i})\}, \tag{3.1}$$

kjer sta $C_{n-1,j}$ in $S_{n-1,k}$ centralni in sekundarni poddel z nivoja $n-1$, $\mu_{n,i}$ in $\sigma_{n,i}$ pa definirata Gaussovo porazdelitev, ki omejuje razlike med lokacijami aktivacij poddelov. Zaradi jasnosti so v naslednjih enačbah indeksi izpuščeni, P, C, S, μ in σ pa predstavljajo kompozicijo in njene parametre.

Posamezna kompozicija je aktivirana (posreduje izhode na višje nivoje), kadar so vsi njeni poddeli aktivirani. Ta pogoj se lahko omehča z mehaniz-

mom halucinacije, opisanem v nadaljevanju. Aktivacija dela je sestavljena iz dveh vrednosti: lokacije L_P, ki predstavlja lokacijo (frekvenco), na kateri se del aktivira in magnitude A_P, ki predstavlja magnitudo aktivacije. Lokacija dela je definirana kot lokacija aktivacije centralnega poddela:

$$L_P = L_C. \tag{3.2}$$

Centralni deli kompozicij na različnih nivojih propagirajo svoje lokacije navzgor po hierarhiji. Magnituda aktivacije je definirana kot:

$$A_P = tanh[G(L_C - L_S, \mu, \sigma) \cdot (A_C + A_S)], \tag{3.3}$$

kjer *tanh* predstavlja hiperbolični tangens, ki omejuje magnitudo med $[0, 1)$, G pa predstavlja Gaussovo funkcijo, ki omejuje razliko v lokacijah centralnega dela in poddelov glede na μ in σ.

Npr. $P_{2,2}$ na sliki 3.1 je definiran kot:

$$P_{2,2} = \{P_{1,1}, P_{1,3}, (1200, 25)\}, \tag{3.4}$$

kjer sta μ in σ podana v centih. Del bo torej aktiviran, kadar bosta $P_{1,1}$ in $P_{1,3}$ na lokacijah približno eno oktavo (1200 centov) narazen. Dve taki aktivaciji sta prikazani na sliki 3.1. Prva na 294 Hz in druga na 440 Hz.

3.1 Relativnost in deljivost delov

Model ima dve pomembni lastnosti, s katerima se razlikuje od podobnih arhitektur. Relativnost delov modelu omogoča, da posamezni del predstavlja abstraktni visokonivojski koncept neodvisno od njegove lokacije v vhodnem signalu.

Zmožnost relativnega zaznavanja izkorišča tudi človeški učni proces. Relativnost tudi minimizira količino pomnilnika, ki je potreben za shranjevanje naučenih konceptov in omogoča ponovno uporabo konceptov v predhodno še ne videnih razmerah.

Relativnost je neločljivo povezana z modelom in je prikazana v definiciji aktivacij delov (enačba 3.3), ki temelji na relativni razdalji aktivacij poddelov. Del je torej lahko aktiviran, kadarkoli so vsi njegovi poddeli aktivirani hkrati na določeni razdalji, ki jo definirata parametra μ in σ. Ker se to lahko zgodi na več lokacijah, ima lahko del več aktivacij na različnih lokacijah. To je prikazano tudi na sliki 3.1, kjer imajo $P_{1,1}$, $P_{1,3}$ in $P_{2,2}$ vsak po dve aktivaciji. To pomeni, da so koncepti, ki jih predstavljajo (v tem primeru množice harmonsko povezanih enot) prisotni na več lokacijah v signalu. Čeprav so deli relativni in sami po sebi predstavljajo samo abstraktne koncepte brez neposrednih absolutnih predstavitev (npr. model ne more izrecno kodirati tona G5, samo koncept tona), aktivacije delov na danih lokacijah kažejo, kje in kdaj se dani koncept pojavi v signalu (npr. del se bo aktiviral na lokaciji, ki ustreza tonu G5). Poleg tega centralni deli v kompozicijah, ki posredujejo svoje lokacije navzgor po hierarhiji, omogočajo analizo aktivacij delov od zgoraj navzdol in pridobivanje frekvenc, ki so jih povzročile.

Relativna narava delov omogoča tudi učinkovito deljenje (ang. shareability) delov. Posamezen del na nivoju L_{n-1} je lahko vsebovan v več kompozicijah na nivoju L_n. Del je torej lahko vključen v več bolj kompleksnih abstrakcij, ki so same po sebi relativne. Npr. del, ki predstavlja koncept tonske višine, je lahko deljen z več kompozicijami na višjem nivoju, ki kodirajo različne intervale. S tem, ko je odpravljena potreba po ohranjanju več absolutnih primerkov absolutnih predstavitev za vsako kompozicijo, lahko model zelo učinkovito kodira kompleksne koncepte.

3.2 Analiza

Aktivacije delov modela je mogoče uporabiti za neposredno interpretacijo vsebine vhodnega zvočnega signala preko aktiviranih konceptov ali kot značilnice za nadaljnjo obdelavo in analizo. Zvočni signal, transformiran v časovno frekvenčno predstavitev, služi kot vhod za nivo L_0. Aktivacije se nato računajo nivo za nivojem, glede na enačbi 3.2 in 3.3. Proces analize je voden z dvema

mehanizmoma: s halucinacijo in inhibicijo. Tretji mehanizem, samodejno uravnavanje jakosti (AGC - automatic gain control), popravlja magnitudo aktivacij v danem časovnem okviru, glede na aktivacije v prejšnjih časovnih okvirih.

Pred definicijo predlaganih mehanizmov je treba predstaviti pojem pokritja. Pokritje $c(P, L_P)$ dela P, aktivno na lokaciji L_P, predstavlja vse informacije signala, pokrite z delom in njegovim poddrevesom delov. Pokritje je izračunano od vrha navzdol od aktivnega dela do nivoja L_0 kot:

$$c(P, L_P) = \bigcup \{c(C, L_P), c(S, L_P + \mu)\}. \tag{3.5}$$

Pokritje dela L_0 je definirano kot množica pozitivnih aktivacij delov $A_P > 0$ in predstavlja množico pokritih frekvenčnih komponent. Primer s slike 3.1: pokritje dela $P_{2,2}$, aktivno na 294 Hz, je množica aktivacij L_0, ki ustreza frekvencam: {294 Hz, 588 Hz, 880 Hz}.

Halucinacija

Generativna zmožnost modela, ki se odraža v strukturi, je razširljiva na proces proizvajanja aktivacij. Poleg neposredno odraženih aktivacij model ustvari dodatne aktivacije z najboljšim prileganjem z zapolnjevanjem manjkajočih ali poškodovanih informacij v glasbenem signalu. To je izvedljivo z omogočanjem aktivacij delov ob prisotnosti manjkajočih aktivacij na nivoju L_0. Manjkajoče informacije v signalu lahko model delno nadomesti na podlagi naučene strukture. V tem primeru model proizvaja aktivacije delov, ki najbolje pokrijejo prisotne informacije. Strukturni fragmenti, ki niso odraženi v aktualnem stanju signala, so halucinirani. Visokonivojska aktivacija dela se torej pojavi, kot bi bila informacija prisotna na nižjih nivojih. S tem lahko model proizvede hipoteze v primerih brez čistih rezultatov.

Vpliv halucinacije je določen s parametrom τ_1, ki je lahko opredeljen za vsak nivo posebej in se lahko med analizo posodablja. Parameter spreminja pogoje, pod katerimi je lahko del aktiviran. Privzeto je aktivacija mogoča le, kadar so vsi njeni poddeli aktivni. S halucinacijo je lahko del P aktiviran

na lokaciji L_P, kadar je število pokritih frekvenčnih komponent $|c(P, L_P)|$, deljeno z največjim številom komponent (poddelov L_0), večje od τ_1. Npr. τ_1 z vrednostjo 0.75 pomeni, da morajo biti 3 izmed 4 komponent v vhodnem signalu prisotne, da se del aktivira.

Z omogočanjem aktivacij ob prisotnosti nepopolnega signala, ustvarjene aktivacije najboljšega prileganja modelu ne omogočijo le polnjenja manjkajočih informacij, temveč ustvarjajo tudi alternativne razlage vhodnega signala. Različni deli modela lahko razložijo iste fragmente informacij v vhodu. Halucinacija vzpodbudi te alternativne predstavitve in omogoča ustvarjanje več hipotez istega vhoda, kar privede do večjega števila robustnih predstavitev signala na višjih nivojih modela.

Inhibicija

Inhibicija izvaja izboljšavo aktivacij z zmanjševanjem števila aktivacij delov na posameznem nivoju. V modelu ponuja ravnotežni dejavnik z zmanjševanjem odvečnih aktivacij, podobno kot stranska inhibicija, ki jo izvaja človeški slušni sistem. Čeprav učni algoritem penalizira aktivacije, ki redundantno pokrivajo signal, se nekateri odvečni deli obdržijo zaradi robustnosti, dodatne odvečne aktivacije pa so še inducirane s halucinacijo.

Aktivacija (L_P, A_P) dela P je inhibirana, kadar drug del (ali množica delov) Q z aktivacijo (L_Q, A_Q) na istem nivoju pokrije iste fragmente informacij v vhodnem signalu, vendar z višjimi aktivacijami. Pogoj je lahko izražen kot:

$$\exists Q : \frac{|c(P, L_P) \backslash c(Q, L_Q)|}{|c(P, L_P)|} < \tau_2 \wedge A_Q > A_P, \tag{3.6}$$

kjer τ_2 definira količino inhibicije. Npr. vrednost 0.5 pomeni, da je aktivacija dela P inhibirana, če je polovica pokritja dela že pokrita z močnejšo aktivacijo drugega dela.

Inhibicija, poleg izboljšave hipotez in odstranjevanja odvečnih aktivacij, zmanjša tudi šum v signalu, ki se po navadi odraža v velikem številu nizkomagnitudnih aktivacij delov na različnih nivojih. Inhibicija in halucinacija

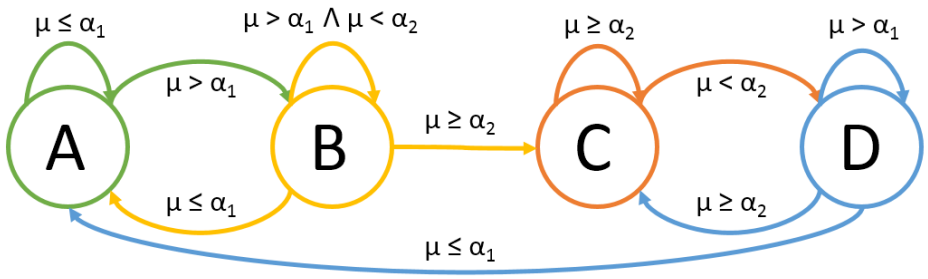
skupaj ponujata učinkovit način za nadzor izrazne moči in robustnosti modela.

Samodejno uravnavanje jakosti

Do zdaj prikazani model je časovno neodvisen. Deluje na osnovi posameznih okvirov, kjer je vsak časovni okvir v časovno frekvenčni predstavitvi obdelan neodvisno od drugih. Zvok pa se s časom razvija in časovno neodvisni modeli pogosto nepravilno odražajo razvoj zvoka.

Model CHM je razširjen s časovno dimenzijo z modeliranjem tako kratkotrajnih kot tudi dolgotrajnih odvisnosti. Za modeliranje kratkotrajnih odvisnosti je uveden mehanizem za kratkotrajno samodejno uravnavanje jakosti AGC (ang. automatic gain control), podoben mehanizmu za samodejno uravnavanje kontrasta v človeških in drugih živalskih zaznavnih sistemih. Mehanizem modelu omogoča povezovanje aktivacij delov skozi čas, tako da je aktivacija v danem trenutku odvisna od prejšnjih aktivacij dela.

Mehanizem AGC spremeni aktivacijo dela na naslednji način: ko se del aktivira na novi lokaciji in aktivacija vztraja, je njena vrednost na začetku vzpodbujena, da poudari vžig in nato zadržana proti stabilni vrednosti (slika 3.3). Delovanje mehanizma AGC, prikazano na sliki 3.2, je definirano s končnim avtomatom s štirimi stanji.

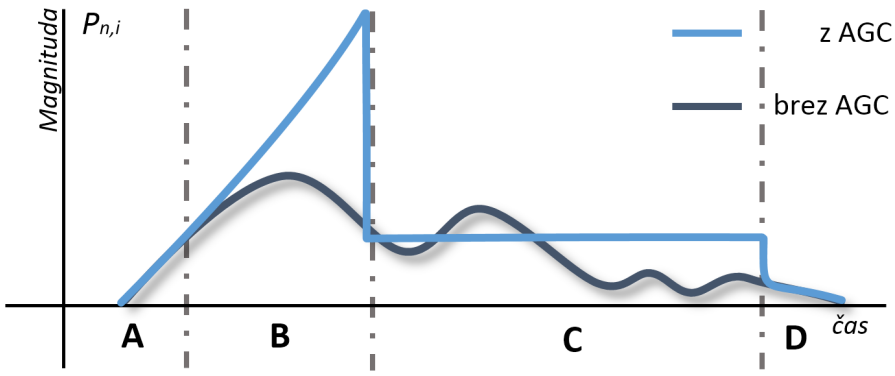


Slika 3.2: Končni avtomat s 4 stanji, ki implementira mehanizem AGC. Stanje A predstavlja običajno delovanje, stanje B predstavlja stanje vzpodbujanja (vžig), stanje C predstavlja zadrževanje in stanje D pojemanje aktivacije. Vir [20]

Parameter μ na sliki 3.2 predstavlja število (v odstotkih) aktivacij dela P na lokaciji L_P v zadnjih N okvirih, α_1 in α_2 pa sta praga za prehode med stanji. Aktivacija dela v posameznih stanjih v času t $A_p(t)$ se izračuna na naslednji način:

$$A_P(t) = \begin{cases} A_P(t): & \text{A, D} \\ \sum_{f=t-N}^{t} A_P(f): & \text{B} \\ s_1: & \text{C} \end{cases},$$

kjer s_1 predstavlja magnitudo aktivacije v zadrževanem stanju (C).



Slika 3.3: Abstraktna predstavitev aktivacij dela skozi čas. Brez mehanizma AGC aktivacija precej niha, še posebej proti koncu dogodka. Samodejno uravnavanje jakosti vzpodbudi vžig dogodka (stanje vzpodbujanja B) in nato zadržuje magnitudo aktivacije (stanje C) na stalni stopnji do konca dogodka (D). Vir [20]

Mehanizem deluje na vseh nivojih. Na nižjih nivojih je učinek kratkotrajen, na višjih nivojih pa dolgotrajen (velikost okna N se povečuje na vsakem nivoju). Poravnan je s kompleksnostjo konceptov predstavljenih na različnih nivojih. Učinek mehanizma je prikazan na sliki 3.3. AGC stabilizira aktivacije, zmanjša šum, ustvari bolj gladek izhod modela in vzpodbuja dogodke vžigov, kot je prikazano na sliki.

3.3 Učenje

Za analizo je potrebno model najprej zgraditi. Gradnja poteka z uporabo nenadzorovanega učenja na množici vhodnih signalov, nivo za nivojem, podobno, kot to počnejo ostale globoke arhitekture. Učni proces uporablja statistike aktivacij delov, ki zajemajo pravilnosti v vhodnih podatkih.

Pri gradnji nivoja L_n se najprej požene analiza vse do nivoja L_{n-1}. Nato se opazujejo sopojavitve aktivacij delov. Za vse pare delov, ki pogosto aktivirajo skupaj na podobnih razdaljah, se tvorijo kompozicije, ki so dodane v množico kandidatov kompozicij P_n. Kadar se tvorijo dvodelne kompozicije, del na nižji lokaciji predstavlja centralni del kompozicije, parametra μ in σ pa sta ocenjena iz množice aktivacij dveh delov, ki se pojavljajo skupaj. Dva ista dela lahko tvorita več različnih kompozicij na različnih razdaljah (kompozicije imajo lahko različne vrednosti parametra μ).

Iz množice vseh kandidatov kompozicij P_n bo na nivo L_n dodana le podmnožica. Postopek izbire je lahko obravnavan kot optimizacijski problem optimalnega pokrivanja informacij v učni množici z minimalno množico kompozicij na nivoju L_n. Išče se torej podmnožica kompozicij $S_n \subset P_n$, ki daje:

- $max(c(S_n))$ - največje pokritje signala po analizi na nivoju L_n,
- $min(S_n)$ - najmanjše število delov na nivoju L_n.

Kriterija zagotavljata zmožnost razlaganja informacij v največji možni meri, hkrati pa model ostane kompakten s čim manj odvečnimi deli (minimizacija redundance).

Izbira delov je implementirana s požrešno metodo, kjer je v vsaki ponovitvi izbrana in na nivo L_n dodana kompozicija iz P_n, ki največ prispeva k pokritju informacij v učni množici. To zagotavlja, da bodo dodani le deli, ki ponujajo dovolj novih informacij glede na trenutno izbrano množico. Izboljšava se ustavi, ko je dosežen eden izmed dveh kriterijev: pokrit je zadosten odstotek informacij v učni množici (glede na prag τ_3) ali pa noben del iz množice kandidatov ne doda zadostnega pokritja informacij (glede na prag τ_4). Algoritem na sliki 3.4 prikazuje opisano metodo.

```
1  procedura IZBOLJSAJ(P_n)
2  prejsnjePokritje ← 0
3  pokritja ← 0
4  L_n ← ∅
5  ponavljaj
6    za vsak P ∈ P_n
7      pokritja[P] ← c(L_n ∪ P)
8    konec
9    izbrano ← argmax(pokritja)
10   L_n ← L_n ∪ izbrano
11   P_n ← P_n \ izbrano
12   ce pokritja[izbrano] − prejsnjePokritje < τ_4 potem
13     prekini // brez dodanega pokritja − zakljuci
14   konec
15   prejsnjePokritje ← pokritja[izbrano]
16 dokler prejsnjePokritje > τ_3 ali P = ∅
```

Slika 3.4: Požrešna metoda za izbiro kompozicij iz množice kandidatov P_n. Prednost imajo deli, ki dodajo največje pokritje informacij v učni množici. Vir [20]

Učenje se začne na nivoju L_1 in se nadaljuje nivo za nivojem, dokler ni doseženo želeno število nivojev, odvisno od uporabe modela. Končni izkupiček vsakega nivoja je množica, ki vsebuje kompozicije delov z dobrim pokritjem učne množice in majhno velikostjo.

Poglavje 4

Nenegativna matrična faktorizacija

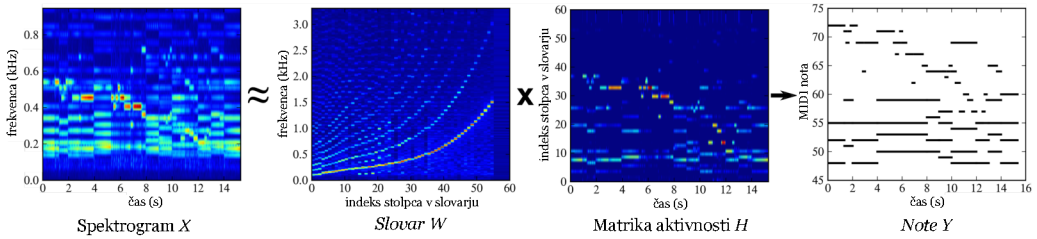
Nenegativna matrična faktorizacija (ang. non-negative matrix factorization) oz. NMF je nenadzorovana tehnika za iskanje predstavitev nad nenegativnimi podatki, ki temeljijo na delih (množica karakterističnih komponent) in se jih lahko aditivno kombinira za rekonstrukcijo opažanj. Če je aplicirana na magnitudni spektrogram polifoničnega signala, lahko NMF najde notne dogodke in pripadajoče aktivnosti, s katerimi lahko optimalno rekonstruira originalni spektrogram.

V splošnem bo pridobljena predstavitev konvergirala v posamezne note, če so izpolnjeni naslednji pogoji. Prvi pogoj zahteva, da mora biti vsak opazovani okvir spektrograma predstavljiv kot nenegativna linearna kombinacija samostojnih notnih spektrov, približek, ki je odvisen od interference med harmoničnimi deli, ki se prekrivajo v polifonični mešanici. Drugi pogoj zahteva linearno neodvisnost osnovnega spektra, tretji pogoj pa zahteva, da so vse note prisotne v bazi.

Zadnji pogoj je težko dosežen v celoti, vendar se v praksi izkaže, da so delne kombinacije dovolj. Aktivnosti, pridobljene z NMF, se izkažejo uporabne kot značilnice za zaznavo posameznih notnih višin, igranih hkrati v danem trenutku v polifoničnem zvočnem signalu. Ta naloga je znana kot

ocenjevanje osnovnih frekvenc in transkripcija zvočnih posnetkov v notni zapis.

NMF je nenadzorovana metoda, zato se jo lahko aplicira na glasbene posnetke brez anotacij. Ker pa so anotacije pogosto prosto dostopne, lahko metoda informacije o tonskih višinah izkoristi za usmerjanje dekompozicij NMF na nadzorovan način za pridobivanje diskriminativnih značilnic, bolj uporabnih za ocenjevanje več višin. Klasifikacija z uporabo NMF v osnovi pomeni izbiranje ene oznake, za ocenjevanje več tonskih višin pa potrebuje več oznak. Boulanger zato uvede DNMF [3], ki predlaga dva diskriminativna kriterija, ki maksimizirata medrazredno razpršenost za vsako oznako posebej in ocenjujeta napovedno moč dane dekompozicije z uporabo logističnih regresorjev. Te ideje so aplicirane v konvencionalnih latentnih spremenljivkah ogrodja NMF in v deterministčnem avtokodirnem modelu za neposredno maksimizacijo diskriminativne zmogljivosti v času testiranja. Za oba modela so oblikovana učinkovita pravila za posodobitve.



Slika 4.1: Ilustracija redke NMF dekompozicije ($\lambda = 0.01$, $\mu = 10^{-5}$) izseka Drigove Serenade. Z uporabo predhodno treniranega slovarja W na polifonični klavirski testni množici je spektrogram X pretvorjen v matriko aktivnosti H, ki je približek klavirske tabulature transkripcije Y. Stolpci matrike W so za namene vizualizacije urejeni po naraščajoči oceni tonske višine. Vir [3]

4.1 Delovanje NMF

Metoda NMF poskuša najti približno faktorizacijo vhodne matrike X:

$$\overset{n_f \times n_t}{X} \simeq \overset{n_f \times n_t}{\Lambda} \equiv \overset{n_f \times m}{W} \cdot \overset{m \times n_t}{H}, \tag{4.1}$$

kjer je X opazovani magnitudni spektrogram s časovno dimenzijo n_t in frekvenčno dimenzijo n_f, Λ je rekonstruirani spektrogram, W je matrika slovarja m osnovnih spektrov, H pa je matrika aktivnosti. Matriki W in H imata nenegativne omejitve $W_{i,j} \geq 0$ in $H_{i,j} \geq 0$. NMF poskuša minimizirati rekonstrukcijsko napako ter deformacijo med opazovanim spektrogramom X in rekonstrukcijo Λ. Pogosta izbira je Evklidska razdalja:

$$C_{LS} \equiv ||X - \Lambda||^2, \tag{4.2}$$

s katero je demonstriran DNMF, čeprav se jo lahko preprosto posploši na druge meritve popačenja v β-divergentni družini. Minimizacijo C_{LS} je lahko dosežena z izmenjevanjem multiplikativnih posodobitev nad H in W:

$$H \leftarrow H \circ \frac{W^T X}{W^T \Lambda}, \tag{4.3}$$

$$W \leftarrow W \circ \frac{X H^T}{\Lambda H^T}, \tag{4.4}$$

kjer operator $\circ$ predstavlja množenje po komponentah, ulomek pa predstavlja deljenje po komponentah. Te posodobitve zajamčeno zmanjšujejo rekonstrukcijsko napako ob predpostavki, da lokalni minimum še ni dosežen. Čeprav je cilj ločeno v W in H konveksen, je skupaj nekonveksen, zato je iskanje globalnega minimuma v splošnem težko rešljivo.

4.1.1 Omejitve redkosti

V polifoničnem signalu z relativno malo sočasnimi toni bi morali biti aktivni elementi $H_{i,j}$ omejeni na majhno podmnožico razpoložljivega osnovnega spektra. Za spodbujanje takega obnašanja se lahko v skupni cilj doda kazen C_S:

$$C_S = \lambda |H|, \tag{4.5}$$

kjer $|\cdot|$ stoji za normo L_1 (največja vsota po stolpcih), λ pa določa relativno pomembnost redkosti. Za odpravitev poddoločenosti, povezane z invarianco WH v transformaciji $W \to WD$, $H \to D^{-1}H$, kjer je D diagonalna matrika, je uvedena omejitev, da mora imeti osnovni spekter enotsko normo. Enačba 4.3 postane:

$$H \leftarrow H \circ \frac{W^T X}{W^T \Lambda + \lambda}, \tag{4.6}$$

Multiplikativna posodobitev za W (enačba 4.4) je zamenjana s projeciranim gradientnim spustom [14]:

$$W \leftarrow W - \mu(\Lambda - X)H^T, \tag{4.7}$$

$$W_{:i} = \frac{W_{:i}}{||W_{:i}||}, \tag{4.8}$$

kjer je $W_{:i}$ i-ti stolpec matrike W, μ stopnja učenja in $1 \leq i \leq m$.

4.2 NMF za ocenjevanje več tonskih višin

Zmožnost pridobivanja osnovnih notnih dogodkov iz polifonične mešanice z uporabo modela NMF je zanimiva za ocenjevanje več tonskih višin. V idealnem scenariju slovar W vsebuje spektralne profile posameznih not, ki sestavljajo kombinacijo in matriko aktivnosti H, ki približno ustreza anotacijam. Primer redke NMF dekompozicije odlomka Drigove Serenade z uporabo predhodno treniranega slovarja na preprosti polifonični klavirski testni množici je ilustriran na sliki 4.1. Slovar vsebuje predvsem monofonični osnovni spekter, ki je zaradi vizualizacije urejen glede na ocenjeno tonsko višino. Opazna je tudi jasna podobnost med matriko aktivnosti in ciljnimi notami v predstavitvi klavirske tabulature Y.

Za ocenjevanje več tonskih višin se lahko izkoristi več možnosti dekompozicije NMF. Pristop preiskovanja slovarja [23] uporablja ocenjevanje prisotnosti tonske višine (ali odsotnost le-te) v vsakem stolpcu matrike W, kar se

lahko naredi samodejno z uporabo harmonskih glavnikov [25] in transkripcijo vseh višin, za katere povezane $H_{i,j}$ aktivnosti presegajo mejo η:

$$Y_{kj} = 1 \Leftrightarrow H_{ij} \geq \eta, \tag{4.9}$$

kjer je $L(i)$ ocenjena oznaka višine (indeks) i-tega osnovnega spektra. Za to metodo se lahko adaptivno izvede nova faktorizacija za vsak analizirani kos ali pa se vnaprej nauči slovar iz razširjenega korpusa in se ga med testiranjem ne spreminja. Slovar je lahko sestavljen tudi z združevanjem izoliranih notnih spektrov.

Druga možnost je ocenjevanje vsakega stolpca matrike Y iz ustreznega stolpca matrike H z uporabo splošnonamenskega klasifikatorja z več oznakami ali z množico binarnih klasifikatorjev, enega za vsako oznako (noto). Za treniranje klasifikatorja je potrebno uporabiti stalni slovar in anotacije not.

4.3 Diskriminativni kriterij

Preprosta interpretacija matrike aktivnosti kot približek transkripcije se po navadi poslabša, če se poveča inštrumentalna raznolikost, višinski razpon ali polifonija. Model DNMF za pridobivanje smiselnih značilnic v W in H uvaja dva diskriminativna kriterija, ki izkoriščata poravnane informacije o notah Y.

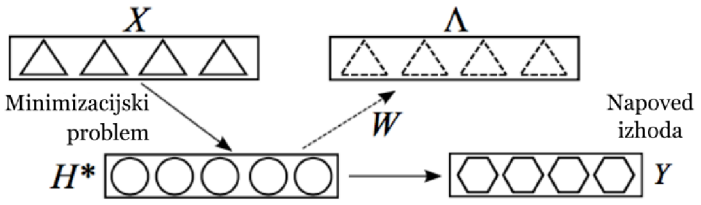
Pri prvem kriteriju želi model maksimizirati medrazredno razpršenost za H_{ij}, kjer se razredi nanašajo na prisotnost oz. odsotnost dane oznake višine ob določenem času. Spodbuja maksimalnost aktivnosti, povezanih z danim osnovnim spektrom, kadar je njihova višina prisotna v notah in minimalna v nasprotnem primeru, tako da je enodimenzionalna odločitvena meja dovoljšna za oceno prisotnosti note.

Najprej se vsakemu stolpcu i v matriki W določi oznako višine $L(i)$ ali nastavi $L(i) = -1$, da se označi osnovni spekter brez tonske višine. Zaradi invariance WH ob stolpčni permutaciji matrike W in enakovredni vrstični permutaciji matrike H, se lahko te dodelitve izvajajo poljubno dolgo, dokler

število osnovnih spektrov, ki opisuje vsako višino q in število spektra $\overline{q}$ brez tonskih višin ostaja nespremenjeno. Bolj natančna oblika tega kriterija je:

$$C_d(H) = \sum_{ij} \begin{cases} -\beta^+ H_{ij} & \text{če } Y_{L(i),j} = 1 \\ \beta^- H_{ij} & \text{če } Y_{L(i),j} = 0 \\ 0 & \text{če } L(i) = -1 \end{cases}, \tag{4.10}$$

kjer β^+ in β^- predstavljata pomembnost prisotnosti in odsotnosti elementov matrike H_{ij}. Limita $\beta^- \to \infty$ ustreza $H_{ij} = 0$ za $Y_{L(i),j} = 0$.



Slika 4.2: V avtokodirnem modelu DNMF je vhod kodiran z deterministčnim minimizacijsim postopkom. Kod $H*$ je treniran za rekonstrukcijo X in predikcijo Y. Vir [3]

Drugi predlagani kriterij ne predpisuje vnaprej določene strukture matrike aktivnosti ampak poskuša ugotoviti, ali je H dober kazalnik za Y. Model vpelje stopnjo logističnega regresorja z matriko uteži V in vektor pristranskosti b z matriko H kot vhodom:

$$p_{kj} = \sigma((VH)_{kj} + b_k), \tag{4.11}$$

kjer je $\sigma(x) \equiv (1+e^{-x})^{-1}$ funkcija logističnega sigmoida po komponentah, p pa izhodna matrika verjetnosti not oz. verjetnostna klavirska tabulatura. Za diskriminativni kriterij matrike H se uporablja križna entropija:

$$C_l(H) = -\alpha \sum_{kj} Y_{kj} log p_{kj} + (1 - Y_{kj}) log(1 - p_{kj}), \tag{4.12}$$

kjer je α koeficient uteži. Z dodajanjem kriterija k skupnemu cilju model DNMF postane:

$$C = C_{LS} + C_S + C_d + C_l. \tag{4.13}$$

Obstaja preprost dokaz, da sta obe Hessovi matriki $\nabla^2_H C_d(H)$ in $\nabla^2_H C_l(H)$ pozitivno semidefinitni in da cilj modela DNMF ostane konveksen ločeno v W in H. Pravilo za multiplikativno posodabljanje matrike H (enačba 4.6) postane:

$$H \leftarrow H \circ \frac{W^T X}{W^T \Lambda + \lambda + \frac{\partial C_d(H)}{\partial H} + \frac{\partial C_l(H)}{\partial H}}, \tag{4.14}$$

kjer so gradienti:

$$\frac{\partial C_d(H)}{\partial H_{ij}} = \begin{cases} -\beta^+ & \text{če } Y_{L(i),j} = 1 \\ \beta^- & \text{če } Y_{L(i),j} = 0 \\ 0 & \text{če } L(i) = -1 \end{cases}, \tag{4.15}$$

$$\frac{\partial C_l(H)}{\partial H} = \alpha V^T (p - Y). \tag{4.16}$$

Pravila za posodobitve matrike W so enaka kot pri redki NMF in so podana z enačbama 4.7 in 4.8. Parametri za V in b so optimizirani s stohastičnim gradientnim spustom z uporabo posodobitev:

$$V \leftarrow V - \mu(p - Y)H^T, \tag{4.17}$$

$$b_k \leftarrow b_k - \mu \sum_j (p_{kj} - Y_{kj}). \tag{4.18}$$

4.4 Avtokodirni model

V verjetnostnem modelu latentnih spremenljivk (ang. latent variables) oz. LV, ki je osnova modela NMF, so aktivnosti obravnavane kot skrite spremenljivke s skupno negativno logaritemsko verjetnostjo podano z enačbo 4.13, uporaba enačb 4.14, 4.7 in 4.8 pa med učenjem ustreza pričakovanjem in maksimizacijskim fazam algoritma EM (expectation-maximization) [13].

Subtilnost, povezana s to interpretacijo, se pojavi v testnih pogojih, kjer matrika Y ni znana. Za pridobitev matrike H bi bila lahko uporabljena enačba 4.6, vendar je lahko za to uporabljen avtokodirni model.

Vrednost matrike H je v testnih pogojih označena kot $H*$:

$$H*(W) \equiv \arg\min_{H}(C_{LS} + C_S). \tag{4.19}$$

Na to spremenljivko sta nato aplicirana diskriminativna kriterija $C_d(H*)$ in $C_l(H*)$. Glede na to, da je $H*$ popolnoma deterministična funkcija vhoda z edinim naučenim parametrom W, lahko se ta model enači z avtokodirnikom, kjer je kodirni korak sestavljen iz kompleksnega minimizacijskega problema (enačba 4.19), dekodirni korak pa je običajna linearna rekonstrukcija vhoda (enačba 4.1). Poleg tega diskriminativni kriteriji spodbujajo, da je $H*$ dober kazalnik matrike Y. Celoten model je prikazan na sliki 4.2. Posodobitev projeciranega gradientnega spusta za W postane:

$$W \leftarrow W - \mu\frac{\partial C(H*)}{\partial W}, \tag{4.20}$$

$$W_{:i} = \frac{W_{:i}}{||W_{:i}||}. \tag{4.21}$$

Ker je $H*(W)$ rezultat optimizacijskega postopka, gradienta $C(H*)$ glede na W ni mogoče preprosto izračunati. Zaradi konvergence multiplikativnih posodobitev (enačba 4.6) je lahko $H*$ izražen kot neskončno zaporedje, skrajšano na K ponovitev:

$$H* = \lim_{k\to\infty} H^k \simeq H^K, \tag{4.22}$$

kjer:

$$H^{k+1} \leftarrow H^k \circ \frac{W^T X}{W^T W H^k + \lambda}. \tag{4.23}$$

Tako se gradienti preprosto izračunajo z vzvratnim širjenjem skozi ponovitev k v učinkovitem času $O(K)$:

$$\frac{\partial C}{\partial H^k} = \frac{\partial C}{\partial H^{k+1}} \circ \frac{H^{k+1}}{H^k} - W^T W B^k, \tag{4.24}$$

za $0 \leq k < K$, kjer je pomožna spremenljivka B^k:

$$B^k = \frac{\partial C}{\partial H^{k+1}} \circ \frac{H^{k+1}}{W^T W H^k + \lambda}. \tag{4.25}$$

Začetni pogoji so:

$$\frac{\partial C}{\partial H^K} = W^T(WH^K - X) + \lambda + \frac{\partial C_d}{\partial H^K} + \frac{\partial C_l}{\partial H^K}. \tag{4.26}$$

Zadnja gradienta sta podana z enačbama 4.15 in 4.16, kjer je $H = H^K$. Gradient glede na W je:

$$\frac{\partial C}{\partial W} = \sum_{k=0}^{K-1} \Big[X\Big(\frac{\partial C}{\partial H^{k+1}} \circ \frac{H^{k+1}}{W^T X}) - W(B^k H^{kT} + H^k B^{kT}\Big)\Big] + (WH^K - X)H^{KT}. \tag{4.27}$$

Ob računanju $\partial C/\partial W$ je natančnost približka končnega zaporedja (enačba 4.22) pomembna samo v bližini trenutne vrednosti W, označena kot W^0. Učinkovitost se lahko povečuje brez žrtvovanja natančnosti z inicializacijo $H^0 \equiv H * (W^0)$ in majhno vrednostjo K (< 10). Ta gradient lahko postane neskončen, kadar je matrika W pomanjkljivo rangirana. To se zgodi, kadar se kombinacije osnovnega spektra začasno poravnajo [9]. Ta optimizacijska težava se v praksi ublaži z dvema dejstvoma: osnovni spekter se ponovno normalizira po vsaki posodobitvi (enačba 4.21), uporaba končnega zaporedja za približek gradienta pa skuša zgladiti singularnosti.

V diplomski nalogi bomo izhod modela CHM uporabili kot učno množico za metodo DNMF in s tem robustnim rezultatom modela CHM izboljšali točnost.

Poglavje 5

Vizualizacija

Razviti želimo orodje za vizualizacijo in analizo glasbenih posnetkov. Za pridobivanje transkripcij želimo uporabiti model CHM, ki nam daje kvalitetne podatke o tonskih višinah. Z uporabo nalaganja zunanjih datotek želimo uporabljati tudi podatke drugih modelov (npr. metoda DNMF). Podatke želimo prikazovati na klavirski tabulaturi, na kateri lahko vidimo spreminjanje tonskih višin skozi čas. Z orodjem želimo predvajati naložene zvočne posnetke in sintetizacijo transkripcij. Podatke na klavirski tabulaturi želimo tudi urejati (dodajanje in brisanje tonov) ter dodajati ritmične anotacije (označevanje taktov in dob).

5.1 Predstavitev glasbe

Z glasbo se po navadi srečamo v akustični obliki, torej s poslušanjem. Če jo želimo posneti, jo v digitalni obliki največkrat zapišemo v obliki pulzno kodne modulacije (ang. pulse code modulation) oz. PCM.

Poleg akustične obstaja tudi simbolična oblika. Ta je največkrat predstavljena z notnim zapisom. Notni zapis je standardiziran in je glasbenikom dobro znan. V digitalni obliki se najpogosteje uporablja format MIDI (Musical Instrument Digital Interface) [15]. Njegova slabost je, da v osnovni obliki predstavlja samo tone in njihove dolžine. S tem izgubimo točne ritmične

informacije, postavitev not, celostno obliko partiture itd. Za notni zapis zato obstajajo drugačni formati. Najbolj podprt je univerzalni format MusicXML [10]. Iz simbolične oblike oz. notnega zapisa lahko akustično obliko dobimo z igranjem glasbenih inštrumentov oz. petjem, samodejno pa s sintezo.

Iz akustične oblike lahko simbolično obliko dobimo z uporabo transkripcije. Transkripcijo lahko delamo ročno, tako, da glasbo poslušamo in zapisujemo note, vendar je ta način precej dolgotrajen. Tukaj si lahko pomagamo s samodejno transkripcijo.

5.1.1 Digitalna akustična oblika

Za digitalno analiziranje akustične glasbe moramo glasbo iz analogne oblike pretvoriti v digitalno z uporabo vzorčenja in kvantizacije. Vzorčenje in kvantizacija pretvorita analogni signal v digitalnega z neko frekvenco vzorčenja in številom bitov. Na glasbo lahko nato gledamo kot na valovno krivuljo (magnituda in čas), frekvenčni spekter (frekvence in magnitude v danem časovnem okviru z uporabo frekvenčne transformacije, npr. Fourierova transformacija) ali spreminjanje frekvenc v času (čas-frekvenca-magnituda imenovano tudi spektrogram, ki ga dobimo z uporabo časovno frekvenčne transformacije). Spektrogram lahko poenostavimo tudi kot klavirsko tabulaturo (ang. piano roll), kjer frekvence prikažemo kot klavirske tone, magnitude pa prikazujemo z različnimi barvami z uporabo barvnih lestvic.

5.2 Spletna storitev

Model CHM je napisan v programskem jeziku C# v okolju .NET. Lahko ga uporabljamo kot knjižnico, za namene vizualizacije pa vsebuje tudi programsko orodje VisualSCH. Orodje VisualSCH za svoje delovanje uporablja Windows Presentation Foundation (WPF), zaradi česar ga lahko uporabljamo le v grafičnem okolju Windows.

Za dostop do modela iz drugih okolij smo model naredili dostopen kot spletno storitev. Ta za svoje delovanje ne potrebuje grafičnega vmesnika, kar

omogoča lažjo prenosljivost. Za vizualizacijo podatkov smo razvili spletno aplikacijo, ki omogoča vizualizacijo modela. Arhitektura aplikacije je več nivojska z naslednjimi sloji:

Spletna aplikacija

Spletna aplikacija je implementirana z uporabno spletnih tehnologij HTML5, CSS3 in JavaScript. Aplikacija uporablja ogrodje AngularJS. Komunicira neposredno s spletno storitvijo preko protokola HTTP.

Spletna storitev

Spletna storitev je aplikacijski programerski vmesnik (API), ki temelji na protokolu REST. Napisana je v programskem jeziku C#. Za prenos podatkov med strežnikom in odjemalci uporablja format JSON preko protokola HTTP. Deluje tudi na platformi Mono, kar pomeni, da jo lahko poganjamo tudi v okoljih Linux in Mac. To nam omogoči tudi preprostejše poganjanje v oblaku.

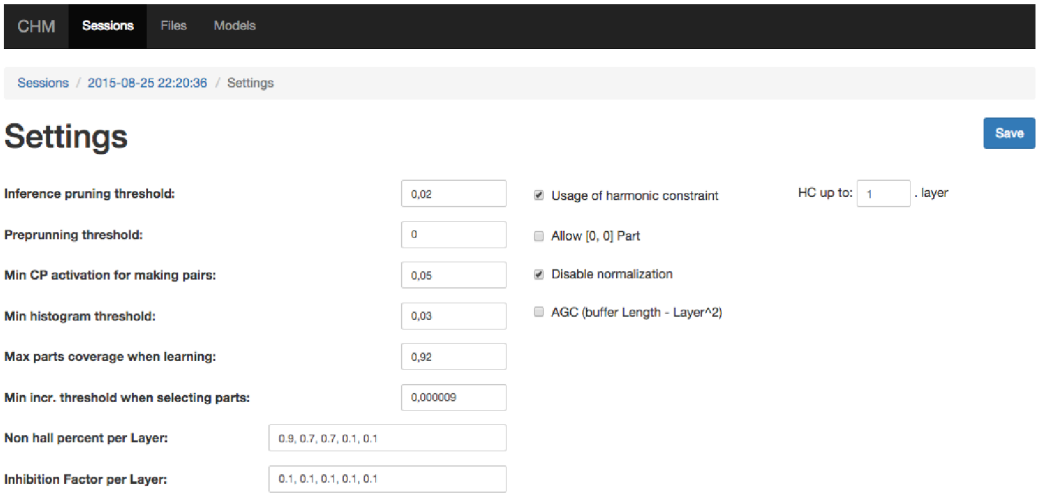
Aplikacijska logika

Aplikacijska logika uporablja vzorec MVVM (Model-View-View-Model). Za pridobivanje podatkov iz glasbenih datotek WAV uporablja knjižnico CHM, za branje podatkov iz datotek MIDI pa knjižnico NAudio.

Hramba podatkov

Aplikacija dostopa do glasbenih datotek (WAV, MAT in MIDI) in bere ter zapisuje naučene strukture modela CHM. Datoteke so shranjene na disku, kar omogoča preprosto upravljanje in arhiviranje.

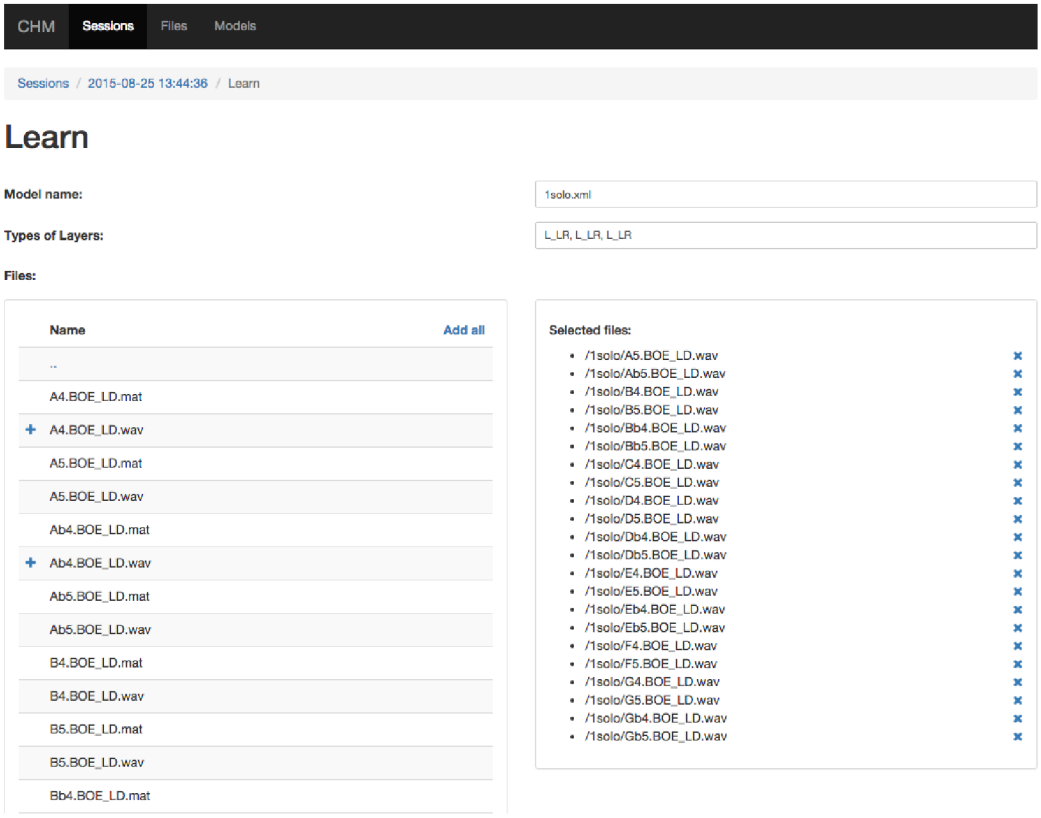
rezultatov metode DNMF). Pridobljene transkripcije lahko predvajamo z vgrajenim predvajalnikom ali jih izvozimo kot MIDI datoteke. Dodajamo lahko tudi anotacije ritmičnih vzorcev.



Slika 5.2: Parametri modela CHM. Nastavljamo lahko parametre modela CHM za učenje in analizo. Nastavimo lahko prag rezanja pri analizi, prag rezanja pred analizo, najmanjšo aktivacijo centralnega dela za generiranje parov, največje pokritje delov pri učenju, najmanjšo spremembo praga pri pokritju za izbiranje delov, odstotek nehaluciniranih delov za posamezni nivo, faktor inhibije za posamezni nivo, normalizacijo in samodejno uravnavanje jakosti (AGC).

5.3.1 Učenje modela CHM

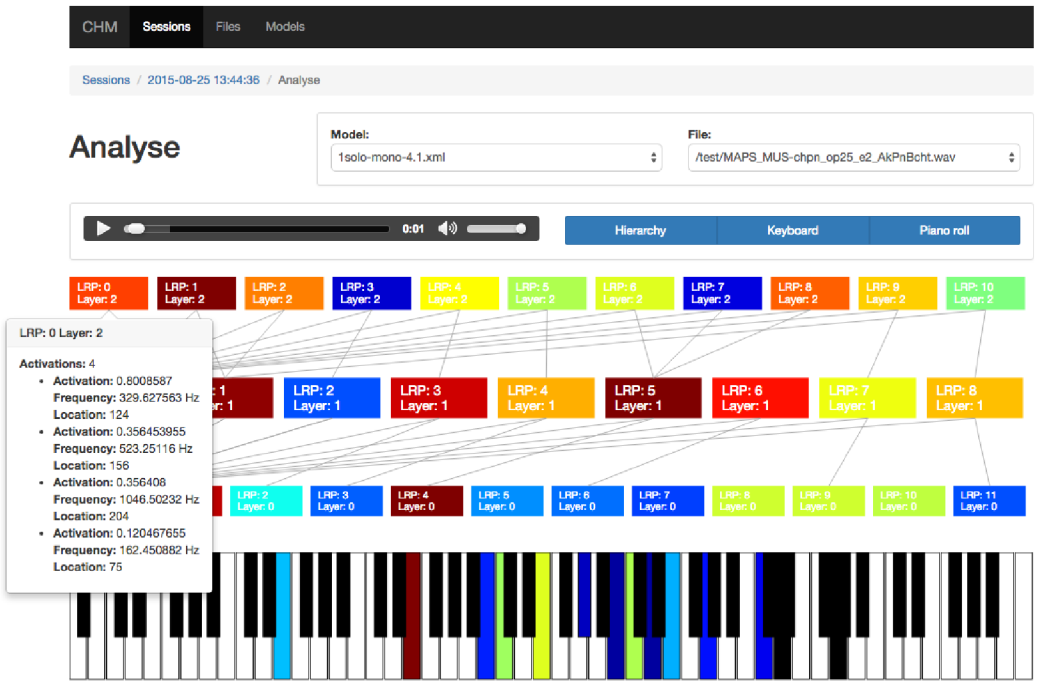
Pri učenju izberemo ime modela, tipe nivojev in datoteke, s katerimi želimo model naučiti. Naučeni model lahko nato uporabljamo za analizo.



Slika 5.3: Učenje modela. Izberemo ime novega modela, tipe nivojev in datoteke, s katerimi želimo model naučiti.

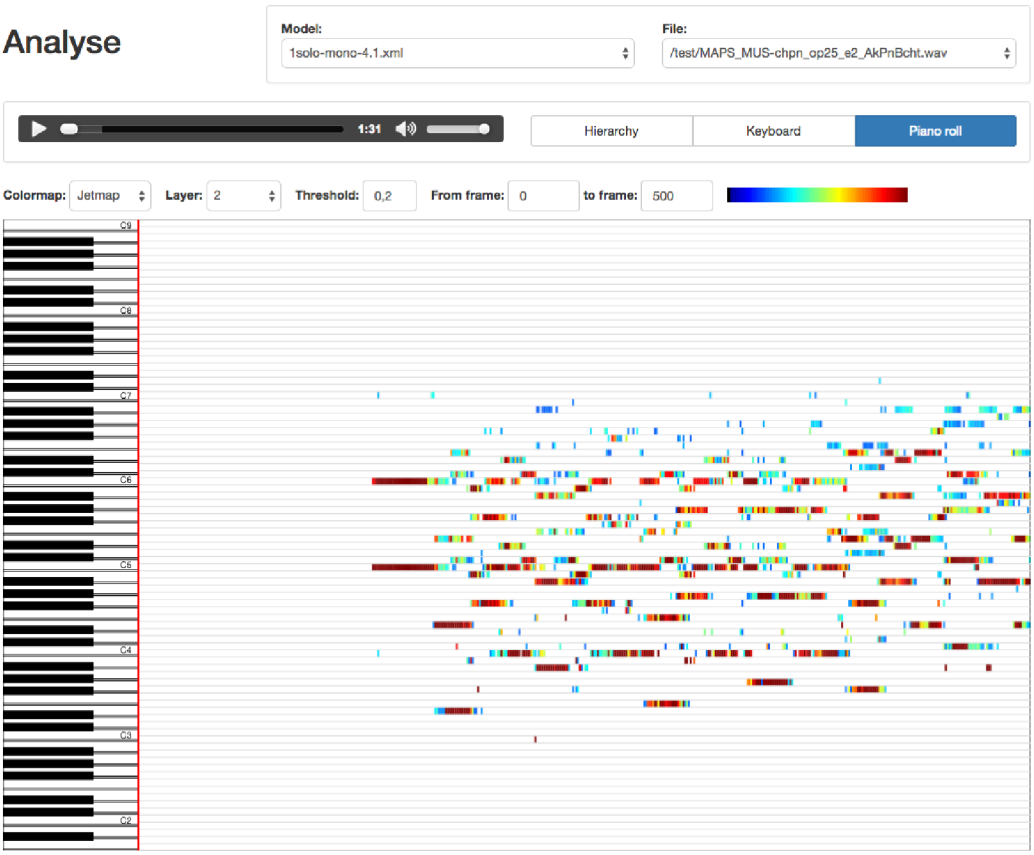
5.3.2 Analiza

Ko model naučimo, lahko z orodjem analiziramo naložene zvočne posnetke. V pogledu hierarhije lahko vidimo naučeno hierarhijo, dele na različnih nivojih, povezave med njimi in podrobne informacije o aktivacijah in njihovih lokacijah. Aktivacije trenutnega okvira lahko vidimo tudi na klaviaturi.



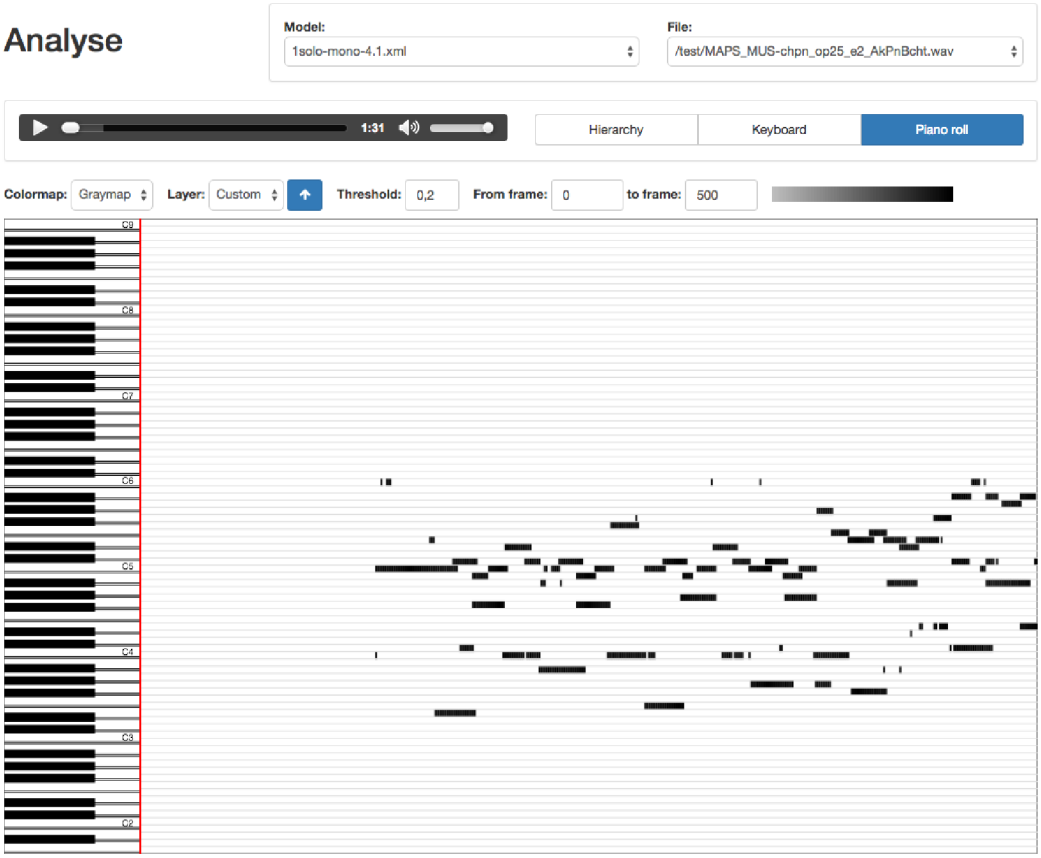
Slika 5.4: Prikaz hierarhije. V pogledu hierarhije lahko vidimo naučeno hierarhijo, dele na različnih nivojih, povezave med njimi in podrobne informacije o aktivacijah in njihovih lokacijah. Aktivacije trenutnega okvira lahko vidimo tudi na klaviaturi.

Naložene zvočne posnetke lahko prikažemo na klavirski tabulaturi. Izbiramo lahko med sivinsko lestvico, lestvico Jet in raznimi enobarvnimi lestvicami. Za podatke iz nivojev modela CHM lahko nastavljamo prag, ki določa, kako močne aktivacije želimo prikazati.



Slika 5.5: Analiza podatkov z drugega nivoja modela. Podatki so prikazani na klavirski tabulaturi. Izbiramo lahko med sivinsko lestvico, lestvico Jet in raznimi enobarvnimi lestvicami. Za podatke iz nivojev modela CHM lahko nastavljamo prag, ki določa, kako močne aktivacije želimo prikazati.

Naložimo in analiziramo lahko tudi podatke, ki smo jih obdelali z drugimi orodji.



Slika 5.6: Prikaz podatkov iz drugih virov. Prikazan je izhod modela DNMF. Podatke smo obdelali s programskim jezikom Python in okoljem Theano, jih izvozili v datoteko JSON in uvozili v našo aplikacijo.

5.3.3 Orodja

Predvajalnik posnetkov

Predvajalnik omogoča predvajanje izbranega posnetka. Na klavirski tabulaturi se med predvajanjem premika kazalec, kar nam olajša analizo s poslušanjem. Z uporabo bližnjic na tipkovnici lahko:

- posnetek predvajamo,
- začasno ustavimo,

- ustavimo in premaknemo na začetek,
- preskakujemo naprej in nazaj,
- predvajamo s polovično hitrostjo,
- predvajamo z normalno hitrostjo ter
- spreminjamo hitrost predvajanja (+/- 1%, +/- 5%).

Klavirska tabulatura

Orodje nam poleg pregleda hierarhije in poslušanja omogoča tudi vizualno analizo in urejanje na klavirski tabulaturi.

Sledi

Klavirska tabulatura omogoča prikazovanje več sledi (ang. tracks). Za vsako sled lahko izberemo barvno lestvico, vir podatkov, prag magnitude in prosojnost. Izbiramo lahko med barvno lestvico Jet (ang. jetmap), sivinsko lestvico (ang. greymap) in različnimi enobarvnimi lestvicami (rdeča, zelena, modra, črna). Za vir podatkov lahko izberemo posamezne nivoje modela CHM, podatke neposredno po predobdelavi (TFM), datoteko MIDI in podatke iz drugih orodij (z nalaganjem datotek JSON). Sled lahko pustimo tudi prazno in jo uporabimo za urejanje. S pragom magnitude nastavimo mejo, nad katero morajo biti vrednosti aktivacije modela CHM, da jih prikažemo. Z uporabo prosojnosti lahko hkrati vidimo več sledi in analiziramo, kje se prekrivajo oz. ujemajo (npr. primerjava rezultatov modela CHM z anotacijami iz datotek MIDI).

Predvajalnik MIDI

Posamezne sledi lahko predvajamo z uporabo vgrajenega predvajalnika datotek MIDI. Iz podatkov posamezne sledi ustvarimo datoteko MIDI, ki jo lahko predvajamo znotraj aplikacije ali pa jo shranimo in uvozimo

v druga orodja (npr. Aria Maestosa[1]). Datoteke ne vsebujejo informacij o tempu in zato niso primerne za neposredno obdelavo v orodjih za urejanje notnih partitur. V vgrajenem predvajalniku lahko izbiramo med različnimi inštrumenti (standard General MIDI[2]).

Pogled valovne oblike

Pod klavirsko tabulaturo je prikaz valovne oblike (ang. waveform). Na tem prikazu lahko spremljamo glasnost posnetka skozi čas.

Povečava

Nastavimo lahko obseg okvirjev (indeks začetnega in končnega okvirja) in s tem približamo ali oddaljimo pogled. Uporabimo lahko tudi bližnjice na tipkovnici (+, -) ali drsno ploščico z uporabo geste ščipanja (ang. pinch gesture) na nekaterih prenosnih računalnikih[3].

Premikanje

Z nastavljanjem obsega okvirjev se lahko po posnetku premikamo. Premikanje je sicer usklajeno s predvajalnikom. Uporabimo lahko bližnjice na tipkovnice (◄, ►) ali drsno ploščico z uporabo geste drsenja z dvema prstoma (ang. horizontal two finger swipe).

Urejanje

Na klavirski tabulaturi lahko podatke urejamo z uporabo različnih orodij. Čarobna palica (ang. magic stick) omogoča prenašanje skupin dogodkov (tonov) na prazno sled. S klikom na dogodek na obstoječi sledi algoritem poišče začetek in konec tona ter na prazno sled doda celoten ton. S čopičem (ang. brush) lahko dodajamo posamezne dogodke oz. z vlečenjem miške barvamo večje površine. Z radirko (ang. eraser) posamezne dogodke odstranjujemo oz. z vlečenjem miške čistimo večja območja. Z uporabo orodja za odstranjevanje celotnih tonov lahko s klikom na posamezni dogodek odstranimo celoten ton.

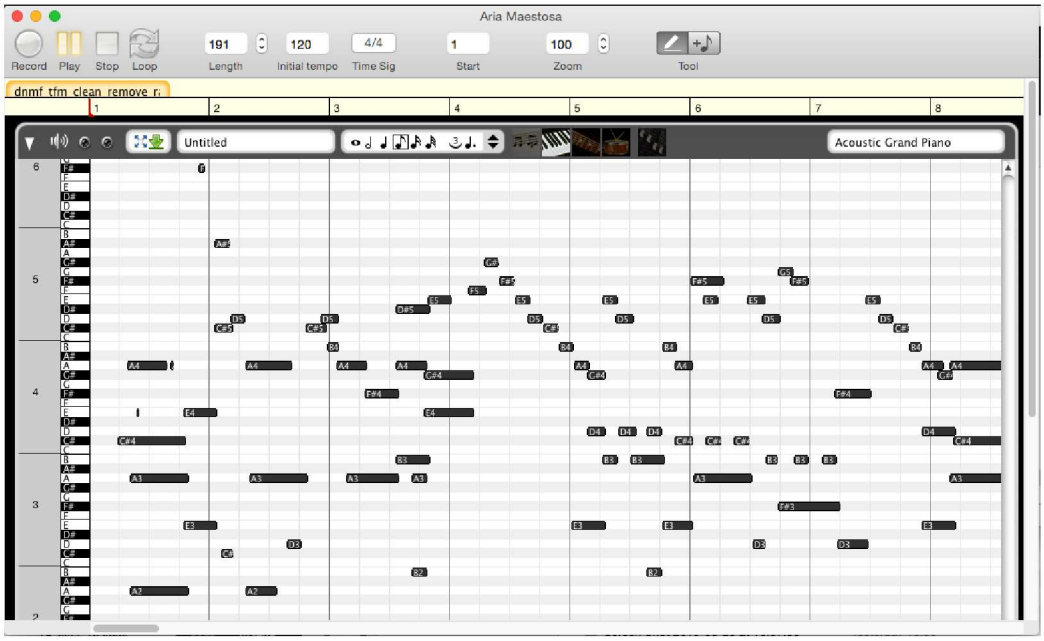
[1] `http://ariamaestosa.sourceforge.net/`

[2] `https://en.wikipedia.org/wiki/General_MIDI`

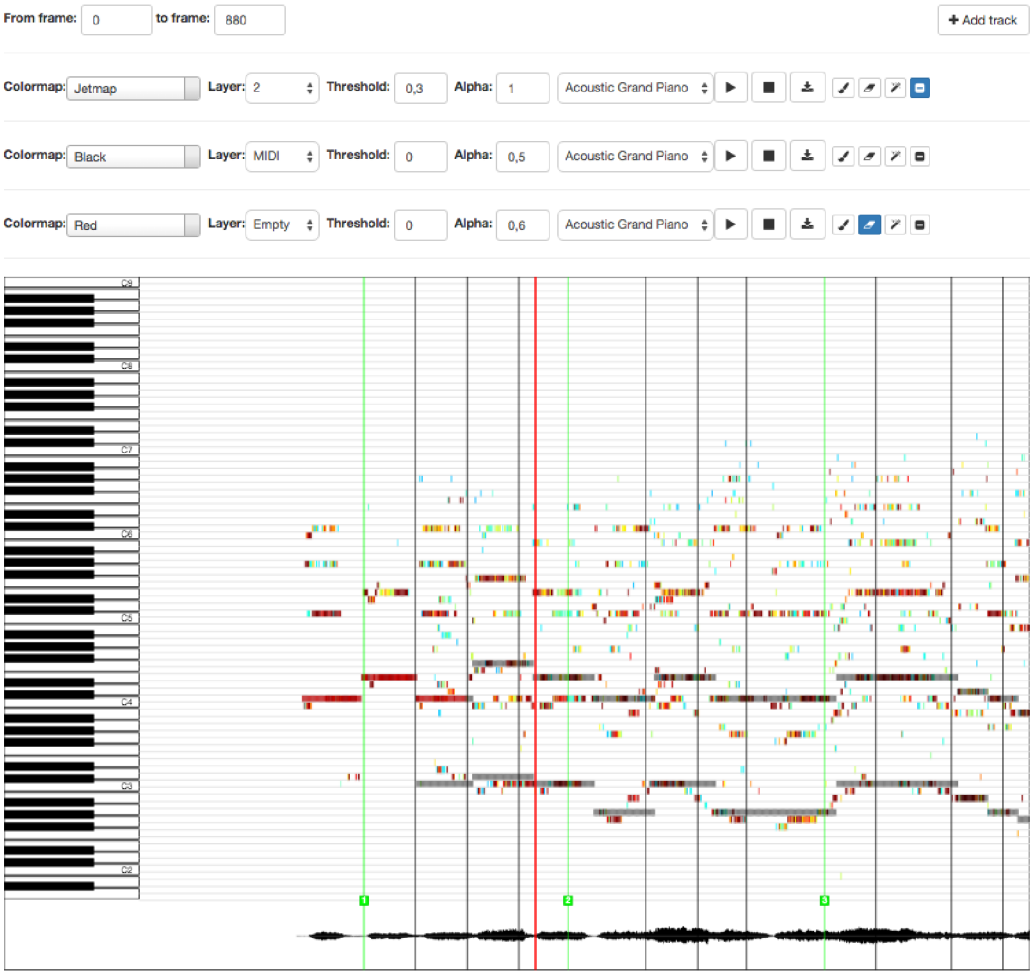
[3] `https://support.apple.com/en-us/HT204895`

Označevanje taktov

Z uporabo bližnjic (tipki M in B) lahko med predvajanjem posnetka na tabulaturi označujemo takte (ang. measures) in udarce (ang. beats), kar nam pomaga pri ročni transkripciji v programih za ustvarjanje notnih partitur, saj tako iz številke takta lažje ugotovimo položaj na posnetku.



Slika 5.7: Prikaz podatkov v aplikaciji Aria Maestosa. Orodje na pogledu klavirske tabulature omogoča izvoz datotek MIDI. Te datoteke lahko uvozimo v druga orodja za nadaljnjo obdelavo.



Slika 5.8: Klavirska tabulatura. Na sliki so prikazani okviri od 0 do 880. Prva sled predstavlja 3. nivo modela CHM z lestvico JET in pragom 0,3. Druga sled predstavlja anotacije iz datoteke MIDI s črno barvo in prosojnostjo 0,5. Na tretjo sled smo z uporabo orodij dodali tri tone. Prikazane so anotacije treh taktov in 9 udarcev. Kazalec je pred drugim taktom. Pod tabulaturo je prikaz valovne oblike.

Poglavje 6

Izboljšava rezultatov

Za izboljšavo natančnosti klasifikatorja smo uvedli različne metode čiščenja.

6.1 Odstranjevanje osamelih dogodkov

Če se dogodek v določenem časovnem okviru ne ponovi dovolj pogosto, ga lahko obravnavamo kot šum in ga odstranimo. Npr. če se v 10 okvirih (100 ms) dogodek pojavi manj kot petkrat, ga lahko odstranimo.

Odstranjevanje osamelih dogodkov formalno definiramo kot filter mediana [19] (ang. median filter). To je filter, ki vrednosti nadomesti s srednjo vrednostjo (mediano) v njihovi okolici.

6.2 Glajenje prekinjenih dogodkov

Če se neki dogodek pojavlja s kratkimi razmiki, lahko to štejemo kot napako in manjkajoča mesta zapolnimo. Npr. če se v 10 okvirih (100 ms) dogodek pojavi več kot petkrat, lahko manjkajoča mesta zapolnimo. S tem tudi podaljšamo dolžino tonov. Tudi glajenje prekinjenih dogodkov formalno definiramo kot filter mediana.

6.3 Odstranjevanje višjih harmonikov

V posnetkih z nizko stopnjo polifonije lahko odstranjujemo dogodke, ki so znotraj istega časovnega okvira od drugih dogodkov oddaljeni z neko razdaljo (npr. za eno oktavo oz. 12 poltonov). Odstranjujemo dogodke, ki imajo pod seboj dogodke (je razlika njihovih tonskih višin enaka določeni razdalji) z višjo magnitudo. Če npr. odstranjujemo oktave (12 poltonov) in v poljubnem časovnem okviru najdemo dogodek s tonsko višino A4 in magnitudo 0.6 ter dogodek z višino A3 in magnitudo 0.9, lahko dogodek z višino A4 odstranimo, ker gre najverjetneje za prvi višji harmonik.

Odstranjevanje višjih harmonikov formalno definiramo kot:

$$A_{P_i}(t) = \begin{cases} 0, & \text{če } \exists A_{P_j}(t) : L_{P_i} = L_{P_j} - C \\ A_{P_i}(t) & \text{sicer} \end{cases}, \tag{6.1}$$

kjer $A_{P_i}(t)$ predstavlja aktivacijo dogodka na nivoju P_i v času t, C predstavlja število centov med lokacijami aktivacij (npr. $C = 1200$ za 12 poltonov), $L_{P_i}(t)$ pa predstavlja lokacijo dogodka na nivoju P_i v času t.

6.4 Odstranjevanje dogodkov izven vokalnega razpona

Pri zbirki slovenskih ljudskih pesmi smo upoštevali vokalni razpon pevcev in odstranili dogodke s tonskimi višinami izven vokalnega razpona.

Odstranjevanje dogodkov izven vokalnega razpona formalno definiramo kot:

$$A_{P_i}(t) = \begin{cases} A_{P_i}(t), & \text{če } L_{P_i} \in [0, C] \\ 0 & \text{sicer} \end{cases}, \tag{6.2}$$

kjer $A_{P_i}(t)$ predstavlja aktivacijo dogodka na nivoju P_i v času t, C predstavlja maksimalno tonsko višino v centih (npr. $C = 6000$ za 60 poltonov oz. 5 oktav), $L_{P_i}(t)$ pa predstavlja lokacijo dogodka na nivoju P_i v času t.

Poglavje 7

Evalvacija

Za primerjavo modela CHM z drugimi modeli na tem raziskovalnem področju smo naredili različne poskuse. Rezultate modela smo primerjali z rezultati drugih raziskovalcev (Boulanger [3], Weninger [26], Marolt [17]). Za natančno primerjavo smo izbrali iste zbirke podatkov in metrike, ki so definirane v [21].

7.1 Ocenjevanje klasifikatorjev

Za ocenjevanje klasifikatorjev uporabljamo različne metrike. Štiri osnovne metrike so:

- resnični pozitivni (TP - true positive) - primer je pozitiven, klasifikator ga je napovedal kot pozitivnega.

  V našem primeru resnični pozitivni predstavljajo pravilno klasificirane tone.

- lažni negativni (FN - false negative) - primer je pozitiven, klasifikator ga je napovedal kot negativnega.

  V našem primeru lažni negativni predstavljajo tone, ki jih nismo klasificirali, vendar se pojavijo v zapisu MIDI.

- resnični negativni (TN - true negative) - primer je negativen, klasifikator ga je napovedal kot negativnega.

V našem primeru resnični negativni predstavljajo vse tone, ki jih nismo klasificirali in se ne pojavijo v zapisu MIDI.

- lažni pozitivni (FP - false positive) - primer je negativen, klasifikator ga je napovedal kot pozitivnega.

  V našem primeru lažni pozitivni predstavljajo napačno klasificirane tone.

7.1.1 Preciznost in priklic

Preciznost (ang. precision) je razmerje med številom pravilno klasificiranih elementov in številom vseh klasificiranih elementov. Priklic (ang. recall) je razmerje med številom pravilno klasificiranih elementov in številom vseh pozitivnih elementov. Vrednosti preciznosti in priklica so med 0 in 1.

$$Preciznost = \frac{\text{število pravilno klasificiranih elementov}}{\text{število vseh klasificiranih elementov}} \tag{7.1}$$

$$Priklic = \frac{\text{število pravilno klasificiranih elementov}}{\text{število vseh pozitivnih elementov}} \tag{7.2}$$

V našem primeru preciznost in priklic za posamezni časovni okvir definiramo kot:

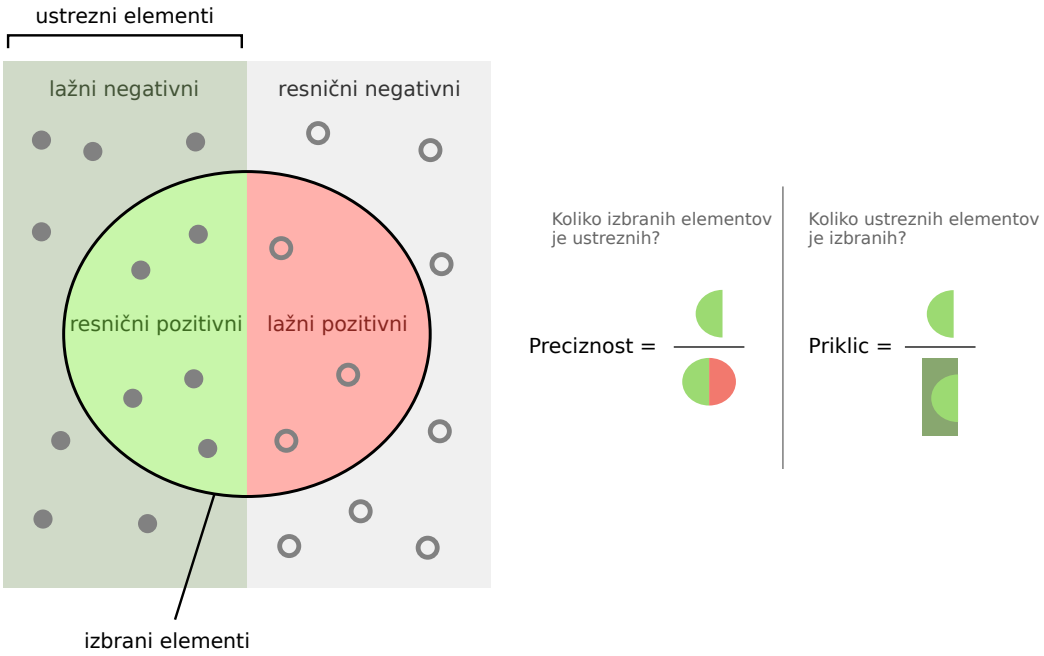
$$Preciznost = \frac{\text{pravilno klasificirani toni}}{\text{vsi klasificirani toni}} \tag{7.3}$$

$$Priklic = \frac{\text{pravilno klasificirani toni}}{\text{vsi toni v zapisu MIDI}} \tag{7.4}$$

7.1.2 Ocena F1

Ocena F1 (ang. F1 score, F-score, F-measure) je uteženo povprečje preciznosti in priklica, njene vrednosti pa so med 0 in 1.

[1] `https://en.wikipedia.org/wiki/Precision_and_recall`



Slika 7.1: Grafični prikaz preciznosti in priklica. Vir Wikipedia[1]

$$F_1 = 2 \cdot \frac{\text{preciznost} \cdot \text{priklic}}{\text{preciznost} + \text{priklic}} \tag{7.5}$$

7.1.3 Natančnost

Natančnost (ang. accuracy) je razmerje med številom ustreznih elementov in številom vseh elementov.

$$\text{Natančnost} = \frac{\text{ustrezni elementi}}{\text{vsi elementi}} \tag{7.6}$$

7.1.4 Natančnost začetkov tonov

Natančnost začetkov tonov (ang. onset accuracy) nam pove, kako točno klasifikator ugotovi začetke tonov. Definicija je enaka natančnosti, vendar gledamo samo okvire, ki v anotacijah vsebujejo začetke tonov.

7.2 Zbirke

Za ocenjevanje točnosti transkripcij obstajajo obsežne zvočne zbirke, ki vsebujejo anotacije (datoteke MIDI). Za naše poskuse smo uporabili klavirsko zbirko MAPS in vokalno zbirko slovenskih ljudskih pesmi.

Zbirka MAPS [6, 7] (MIDI Aligned Piano Sounds) je zbirka klavirskih posnetkov za ocenjevanje več tonskih višin in samodejno transkripcijo glasbe. Vsebuje 31 GB visokokvalitetnih zvočnih posnetkov v obliki WAV. Posnetki so bili ustvarjeni z uporabo navideznih klavirjev in pravega klavirja Yamaha Disklavier. Posnetih je devet kombinacij različnih klavirjev in snemalnih pogojev. Zbirka poleg posnetkov vsebuje tudi anotacije v obliki MIDI in tekstovnih datotek. Posnetki so bili, za zagotavljanje čim večje natančnosti, ustvarjeni iz anotacij. Zbirka je prosto dostopna pod licenco Creative Commons. Vsebuje posnetke posameznih tonov različnih glasnosti, kombinacije intervalov in posnetke celotnih skladb. Posnetki so razdeljeni v več map, ki predstavljajo variacije različnih modelov klavirja in snemalnih pogojev.

Zbirka slovenskih ljudskih pesmi vsebuje 37 anotiranih posnetkov. Pesmi so bile posnete v vsakdanjem okolju z amaterskimi pevci in prenosno snemalno opremo.

7.3 Testiranje

Za preverjanje točnosti klasifikatorja modela CHM smo izvedli več poskusov. Uporabili smo različne zbirke glasbenih posnetkov. S predobdelavo (konstantna Q transformacija in iskanje vrhov) smo iz zvočnih posnetkov dobili podatke v obliki klavirske tabulature. Model CHM smo zgradili z učenjem na posnetkih posameznih klavirskih tipk in na posnetkih enoglasnih vokalnih pesmi. Posamezne pesmi smo nato z naučenim modelom obdelali in rezultate primerjali z anotacijami (datoteke MIDI). Izračunali smo preciznost (P), priklic (R), oceno F1 ter natančnost začetkov tonov (accuracy). Pridobljene rezultate smo nato primerjali z rezultati drugih orodij (npr. *SONIC*[16] in *DNMF-AE* v kombinaciji s *SVM*[3]).

Mapa	Model inštrumenta	Snemalni pogoji	Inštrument ali programska oprema
StbgTGd2	Hybrid	privzete nastavitve	The Grand 2 (Steinberg)
AkPnBsdf	Boesendorfer 290 Imperial	cerkev	Akoustik Piano (Native Instruments)
AkPnBcht	Bechstein D 280	koncertna dvorana	Akoustik Piano (Native Instruments)
AkPnCGdD	Concert Grand D	studio	Akoustik Piano (Native Instruments)
AkPnStgb	Steingraeber 130 (upright)	jazz klub	Akoustik Piano (Native Instruments)
SptkBGAm	Steinway D	ambient	The Black Grand (Sampletekk)
SptkBGCl	Steinway D	zaprt prostor	The Black Grand (Sampletekk)
ENSTDkAm	Yamaha Disklavier Mark III (upright)	ambient	pravi klavir (Disklavier)
ENSTDkCl	Yamaha Disklavier Mark III (upright)	zaprt prostor	pravi klavir (Disklavier)

Tabela 7.1: Mape v zbirka MAPS. Zbirka vsebuje 3 različne navidezne klavirje v različnih snemalnih pogojih in eden pravi klavir v dveh pogojih. Vir [6, 7]

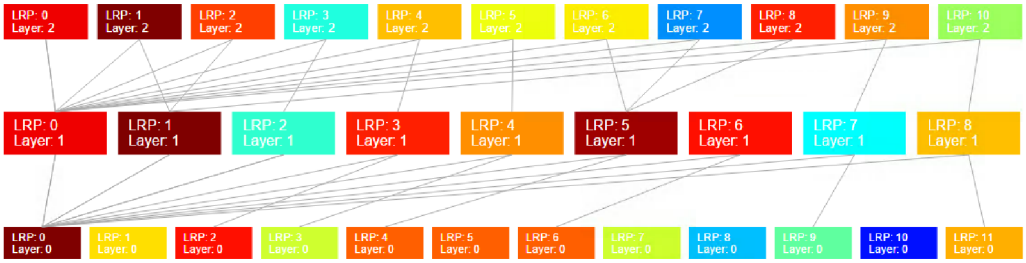
Za poganjanje poskusov smo uporablili programsko okolje Python. S spletno storitvijo smo komunicirali preko vmesnika REST (prikazano na sliki 5.1). Uporabili smo knjižnice Requests, NumPy, SciPy, SciKit Learn in Theano. V Pythonu smo uporabili tudi metodo DNMF, kjer smo podatke iz modela CHM uporabili kot vhod metode DNMF za učenje in evalvacijo.

7.3.1 Predobdelava

Pred uporabo testnih posnetkov v modelu CHM smo vse zvočne posnetke (datoteke WAV) obdelali s konstantno Q transformacijo [4] in poiskali vrhove. Frekvenčni spekter smo razdelili na 345 kanalov med 55 in 8000 Hz s časovnimi okviri 10 ms.

7.3.2 Učenje modela CHM

Za evalvacijo zbirke MAPS smo model CHM zgradili z uporabo 24 posnetkov klavirskih tonov med C4 in B5. Naučili smo 3 nivoje, kjer prvi nivo vsebuje 12 delov, drugi 9 in tretji nivo 11 delov.



Slika 7.2: Model CHM naučen na 24 klavirskih tonih.

Pri zbirki slovenskih ljudskih pesmi je bila gradnja modela otežena, saj zbirka ne vsebuje posnetkov posameznih tonov. Za učenje strukture modela smo uporabili 4 pesmi s stopnjo polifonije 1. Prvi nivo modela vsebuje 12 delov, drugi 10 in tretji nivo 14 delov.

7.3.3 Pridobivanje podatkov iz modela CHM

Iz aktivacij delov na različnih nivojih lahko preko lokacij delov pridemo do frekvenčnih kanalov. Frekvenčne kanale smo za nadaljnjo obdelavo razdelili na 88 tonov (klavirska tabulatura). Za vsak ton smo izračunali vsoto aktivacij, jih normalizirali in tako dobili magnitude.

7.3.4 Ocenjevanje rezultatov

Za ocenjevanje modela smo uporabili preciznost (P), priklic (R), oceno F1 in natančnost začetkov tonov (accuracy). Preciznost, priklic in oceno F1 smo računali za vsak okvir (10 ms), za natančnost začetkov tonov pa smo uporabili okno 100 ms.

7.3.5 Izboljšava modela z uporabo DNMF

Za izboljšavo rezultatov modela CHM smo uporabili metodo DNMF [3] z uporabo programske zbirke Theano [2]. Metoda DNMF se lahko z uporabo učne množice in množice za validacijo zelo dobro prilagodi podatkovni zbirki. Za učenje smo potrebovali tudi anotacije.

7.3.6 Izboljšava rezultatov z uporabo čiščenja podatkov

Za izboljšavo rezultatov modela CHM na zbirki slovenskih ljudskih pesmi smo uporabili metode za čiščenje podatkov, opisane v poglavju 6. Metode smo večkrat ponovili v različnem vrstnem redu.

Primer čiščenja posnetka iz zbirke slovenskih ljudskih pesmi

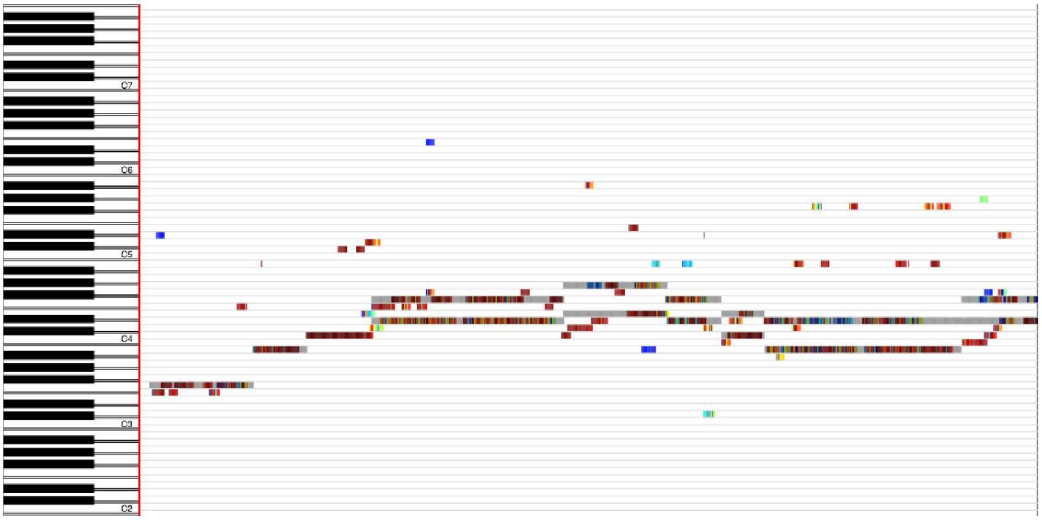
Slika 7.3: Neposredni podatki iz modela CHM. Ocena F1: 0.37

Razdalja	**Ocena F1**
11	0.38
12	0.45
13	0.47
16	0.48
17	0.49
18	0.49
19	0.52
24	0.53
26	0.53
28	0.53

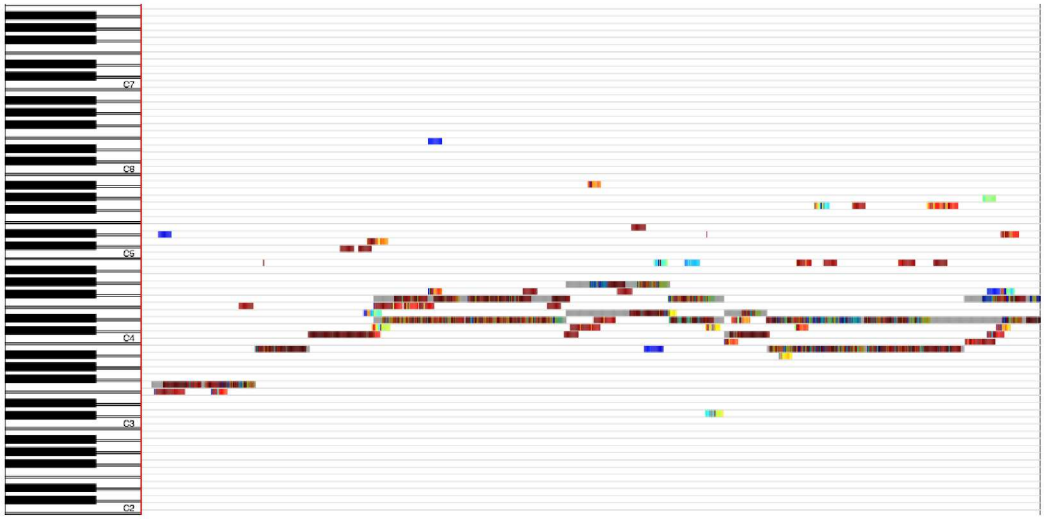
Tabela 7.2: Sprememba ocene F1 z odstranjevanjem višjih harmonikov. Razdalja predstavlja število poltonov od osnovnega dogodka.

Slika 7.4: Odstranjevanje višjih harmonikov. Ocena F1: 0.53

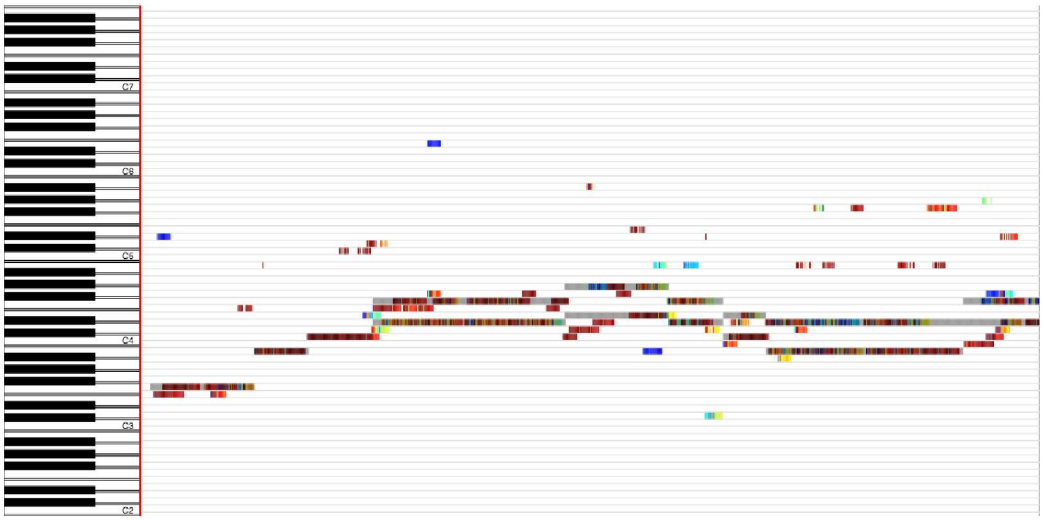
Slika 7.5: Glajenje prekinjenih dogodkov. Ocena F1: 0.58



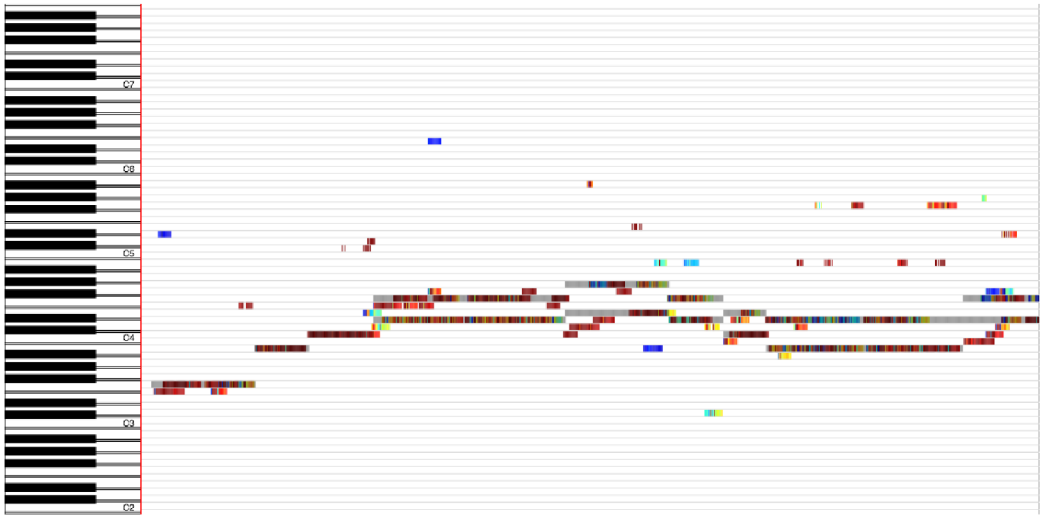
Slika 7.6: Odstranjevanje osamelih dogodkov. Ocena F1: 0.66



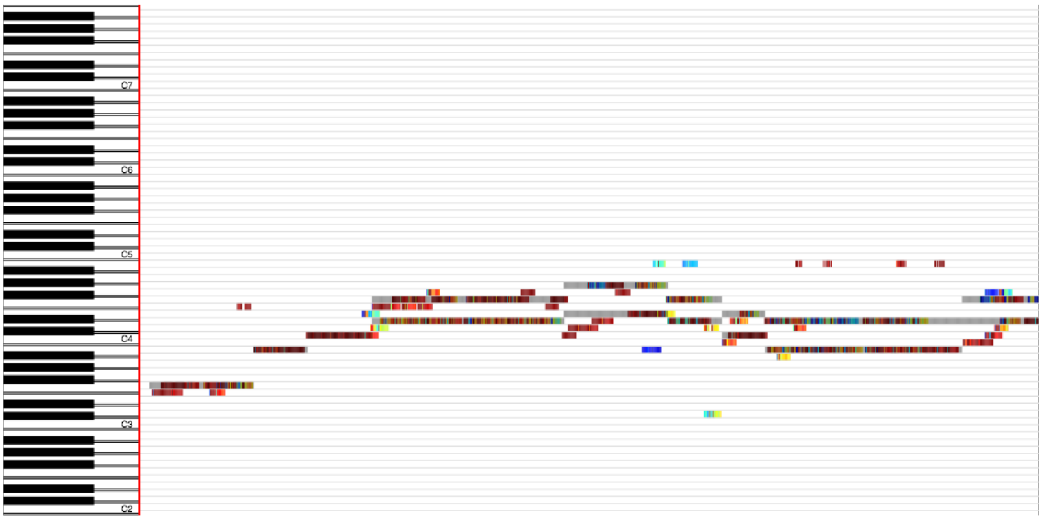
Slika 7.7: Ponovno glajenje prekinjenih dogodkov. Ocena F1: 0.66



Slika 7.8: Ponovno odstranjevanje višjih harmonikov. Ocena F1: 0.67



Slika 7.9: Ponovno odstranjevanje osamelih dogodkov. Ocena F1: 0.68



Slika 7.10: Odstranjevanje dogodkov izven vokalnega razpona. Ocena F1: 0.70

7.3.7 Poskusi

Posamezne mape v zbirki MAPS

Za prikaz točnosti modela CHM na klavirskih posnetkih smo uporabili zbirko MAPS. Zbirka vsebuje več map, ki predstavljajo različne prave in navidezne

klavirje v različnih snemalnih okoljih.

Primerjamo rezultate različnih nivojev modela CHM in modela DNMF. Za učenje modela DNMF smo v vsaki mapi za učno množico vzeli 70% posnetkov. Model CHM in DNMF smo nato testirali na preostalih 30%. Za učenje modela DNMF smo uporabili podatke neposredno po predobdelavi (TFM) ter 1. in 3. nivoja modela CHM.

	CHM 1		CHM 3		DNMF TFM		DNMF CHM 1		DNMF CHM 3	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
AkPnBcht	38.0	40.3	40.4	48.5	65.7	67.4	54.2	60.8	42.8	52.3
AkPnBsdf	39.5	41.7	41.5	47.8	55.9	57.1	51.6	56.2	44.1	51.9
AkPnCGdD	40.1	44.4	43.2	49.6	63.1	66.0	56.2	61.9	46.8	53.3
AkPnStgb	45.2	50.7	44.7	56.5	59.3	63.6	55.2	62.0	46.0	57.6
ENSTDkAm	41.3	38.7	40.3	41.8	54.9	49.2	50.4	47.5	42.2	43.6
ENSTDkCl	37.9	38.6	37.0	42.1	56.0	54.7	45.7	47.0	39.6	45.0
SptkBGAm	42.5	44.7	44.9	52.0	62.2	63.5	58.2	62.3	47.5	54.4
SptkBGCl	45.2	49.0	44.8	53.6	62.6	64.6	59.4	63.6	48.4	55.6
StbgTGd2	43.8	45.8	42.1	48.2	62.3	61.4	51.7	55.2	43.0	48.6

Tabela 7.3: Posamezne mape v zbirki MAPS. *CHM 1* in *CHM 3* predstavljata 1. in 3. nivo modela CHM, *DNMF TFM* predstavlja podatke neposredno po predobdelavi, *DNMF CHM 1* in *DNMF CHM 3* pa 1. in 3. nivo modela CHM kot vhod v model DNMF. *F1* predstavlja oceno F1, *Acc* predstavlja natančnost začetkov tonov.

Učenje na navideznih klavirjih in testiranje na klavirju Yamaha Disklavier

Za prikaz robustnosti modela CHM smo model DNMF učili na posnetkih navideznih klavirjev in testirali na klavirju Yamaha Disklavier.

Primerjamo rezultate različnih nivojev modela CHM in modela DNMF na posnetkih klavirja Yamaha Disklavier. Za testiranje smo uporabili mapi *ENSTDkAm* in *ENSTDkCl*, za učenje modela DNMF pa vse preostale mape. Za učenje modela DNMF smo uporabili podatke neposredno po predobdelavi (TFM) ter 1. in 3. modela CHM. Analizirali smo prvih 30 sekund vsakega posnetka in za primerjavo dodali še rezultate orodja *SONIC*[16] in *DNMF-AE* v kombinaciji s *SVM*[3] z uporabo predobdelave opisane v [25].

	P	R	F1	P_{50}	R_{50}	$F1_{50}$	Acc_{50}
CHM 1	53.5	37.6	42.4	60.8	51.4	53.8	44.1
CHM 3	40.6	55.6	43.8	45.5	68.8	51.6	39.8
DNMF TFM	58.0	56.3	56.2	62.8	69.4	65.0	57.3
DNMF CHM 1	63.0	45.4	50.9	69.5	61.5	63.3	54.2
DNMF CHM 3	59.7	38.6	44.9	68.8	59.4	61.2	50.4
SONIC	61.0	57.1	57.4	64.4	66.8	63.7	50.5
DNMF-AE SVM	/	/	/	66.8	68.7	67.8	/

Tabela 7.4: Yamaha Disklavier v zbirki MAPS. *CHM 1* in *CHM 3* predstavljata 1. in 3. nivo modela CHM, *DNMF TFM* predstavlja podatke neposredno po predobdelavi, *DNMF CHM 1* in *DNMF CHM 3* pa 1. in 3. nivo modela CHM kot vhod v model DNMF. *SONIC* predstavlja orodje SONIC[16], *DNMF-AE SVM* pa predstavlja *DNMF-AE* v kombinaciji s *SVM*[3] z uporabo predobdelave opisane v [25].

Preizkus robustnosti z različnimi inštrumenti

Za preizkus robustnosti smo na prvi mapi zbirke MAPS (*AkPnBcht*) testirali različne inštrumente. Z uporabo datotek MIDI in zvočne pisave (ang. soundfont) FluidR3GM2[2] smo sintetizirali posnetke kitare (2^3), flavte (74), orgel (20), violine (41) in klavirja (1). Model CHM smo naučili na posnetkih klavirskih tipk, za učno množico modela DNMF pa smo vzeli 70% mape. Testirali smo na preostalih 30% posnetkov. Uporabili smo 1. in 3. nivo modela CHM, za vhod v model DNMF pa smo uporabili neposredne podatke po predobdelavi (*TFM*) ter 1. in 3. nivo modela CHM.

[2] `https://musescore.org/en/handbook/soundfont`
[3] `https://en.wikipedia.org/wiki/General_MIDI`

	CHM 1	CHM 3	DNMF TFM	DNMF CHM 1	DNMF CHM 3
Klavir	37.7	47.2	**66.5**	56.8	47.5
Violina	26.8	34.0	**38.3**	38.2	36.3
Flavta	35.2	**44.5**	39.2	39.4	43.4
Orgle	16.1	32.5	25.0	31.5	**35.5**
Kitara	38.7	44.7	35.3	38.7	**44.8**

Tabela 7.5: Preizkus robustnosti z različnimi inštrumenti. *CHM 1* in *CHM 3* predstavljata 1. in 3. nivo naučenega modela CHM, *DNMF TFM* predstavlja podatke neposredno po predobdelavi, *DNMF CHM 1* in *DNMF CHM 3* pa 1. in 3. nivo modela CHM kot vhod v model DNMF. Najboljši rezultati za posamezni inštrument so odebeljeni.

Zbirka slovenskih ljudskih pesmi

Robustnost modela smo preizkusili tudi na vokalni zbirki slovenskih ljudskih pesmi, ki vsebuje 37 večglasnih vokalnih pesmi. Strukturo modela CHM smo zgradili z uporabo 4 pesmi s stopnjo polifonije 1. Za učenje modela DNMF smo uporabili 70% zbirke, 30% pa smo uporabili za testiranje. Za primerjavo smo zbirko testirali tudi z uporabo metode *multiF0*[11] in orodja *SONIC*[16].

	Precision	Recall	F1
CHM 3	**49.6**	**49.5**	**49.3**
DNMF CHM 1	46.6	33.6	38.7
DNMF TFM	41.5	31.1	35.0
multiF0	43.2	47.5	44.8
SONIC	32.6	47.4	37.6

Tabela 7.6: Preizkus zbirke slovenskih ljudskih pesmi. *CHM 3* predstavlja 3. nivo modela CHM, *DNMF CHM 1* predstavlja 1. nivo modela CHM kot vhod v model DNMF, *DNMF TFM* predstavlja podatke neposredno po predobdelavi, *multiF0* predstavlja metodo *multiF0*[11], *SONIC* pa predstavlja orodje *SONIC*[16]. Za testiranje z modelom DNMF je uporabljenih samo 30% zbirke (70% smo uporabili za učenje). Odebeljeni rezultati so najboljši ob primerjavi med pristopi.

7.4 Analiza in diskusija

Zbirka MAPS vsebuje klavirske posnetke, ki so ali sintetizirani ali posneti v prilagojenem studijskem okolju. Zvok je zelo čist, monoton in brez šuma. Zaradi velikega števila posnetkov, dobre kakovosti in dolgega učenja se lahko model DNMF učni množici zelo dobro prilagodi. Za učenje uporabimo kar 70% posnetkov in zelo verjetno je, da se vzorci v preostalih 30% začnejo ponavljati, kar pripelje do tako dobrih rezultatov. Model CHM se uči samo na posnetkih posameznih tonov in se nauči samo strukture oz. definicije posameznih tonov (razmerja med frekvencami oz. harmoniki), model DNMF pa učimo na polifoničnih posnetkih različnih melodij. Zato na podatkih neposredno po predobdelavi z modelom DNMF dobimo precej boljše rezultate kot neposredno z modelom CHM, vendar pa je model CHM bolj robusten, kar vidimo v poskusu z različnimi inštrumenti. Harmonske strukture, ki so rezultat ekstrakcije skupnih lastnosti zahodnih inštrumentov, so si precej podobne, zato z modelom CHM na višjih nivojih dobimo dobre rezultate. Ker so rezultati višjih nivojev dovolj konsistentni, jih lahko z modelom DNMF

še dodatno izboljšamo. V zbirki slovenskih ljudskih pesmi smo še dodatno pokazali robustnost modela. Tukaj so snemalni pogoji zelo slabi, zvočni posnetki pa nedosledni, zato nam učenje harmonskih struktur vokalnega petja prinaša boljše rezultate kot učenje polifoničnih vzorcev. Primerjava z drugimi rešitvami pokaže, da so rezultati modela CHM primerljivi oz. pri zbirki slovenskih ljudskih pesmi celo boljši.

Poglavje 8

Zaključek

Uspešno smo razvili orodje za analizo in urejanje transkripcij z uporabo modela CHM. Aplikacija je prenosljiva, saj deluje v vseh novejših brskalnikih, ki podpirajo standard HTML5, hkrati pa jo lahko uporablja več uporabnikov. Z različnimi poskusi smo pokazali visoko klasifikacijsko točnost in robustnost modela CHM. Dodatno lahko z ročnim in samodejnim pametnim čiščenjem podatkov točnost še izboljšamo.

8.1 Prednosti in omejitve orodja

Pri razvoju orodja je bilo veliko poudarka na praktični uporabi. Orodje nam v praksi pomaga pri izdelavi transkripcij in z nekaj ročnimi popravki dobimo uporabne rezultate. Prednost modela je tudi interaktivna uporaba, saj lahko ob poslušanju posnetkov rezultate sproti popravljamo. Pomembna je tudi hitrost analize. Posnetke lahko naložimo kar preko spletnega brskalnika in v nekaj minutah dobimo rezultate. Orodje nas ne omejuje samo na model CHM, saj lahko v aplikacijo uvozimo tudi rezultate drugih algoritmov in orodje uporabljamo za analizo in urejanje zunanjih podatkov.

Model trenutno deluje na principu časovnih okvirov in ne pozna koncepta dolžine tonov. Za izvoz v datoteke MIDI daljše tone sestavljamo iz zaporedij tonskih višin v zaporednih časovnih okvirih kar morebitne krajše tone

združuje v daljše. Prav tako orodje ne pozna koncepta tempa oz. taktnih načinov, kar predstavlja težavo pri uvozu v orodja za notacijo, ki delujejo na principu notnega črtovja in taktnih načinov.

8.2 Nadaljnje delo

Trenutno orodje uporabljamo le sami, želimo pa ga ponuditi tudi drugim. Predobdelava posnetkov in analiza z modelom CHM se trenutno izvaja na strežnikih, potrebuje pa kar veliko računskih zmogljivosti. Če želimo orodje ponuditi zunanjim uporabnikom, bi morali uporabo strežniških virov zaračunati, kar pa omejuje uporabnost, zato želimo obdelavo posnetkov prenesti v brskalnik. Tako ne bomo več omejeni z lastnimi strežniškimi viri.

Dodati želimo tudi možnost ustvarjanja uporabniških računov in deljenja posnetkov ter rezultatov transkripcije. Spodbujati želimo tudi uporabo drugih algoritmov in primerjanje rezultatov.

Izboljšati želimo uporabniško izkušnjo in videz aplikacije ter dodati nove funkcionalnosti. Z uporabo označevanja taktov bi lahko podatke poravnali in ustvarili datoteke MIDI, ki vsebujejo informacije o času, kar bi nam poenostavilo uvoz v druga orodja.

Literatura

[1] Samer A. Abdallah and Mark D. Plumbley. Polyphonic transcription by non-negative sparse coding of power spectra. In *Proceedings of the 5th International Conference on Music Information Retrieval*, Barcelona, Spain, October 10-14 2004. `http://ismir2004.ismir.net/proceedings/p058-page-318-paper216.pdf`.

[2] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: A cpu and gpu math compiler in python. In *Proc. 9th Python in Science Conf*, pages 1–7, 2010.

[3] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Discriminative non-negative matrix factorization for multiple pitch estimation. In *ISMIR*, pages 205–210, 2012.

[4] Judith C Brown and Miller S Puckette. An efficient algorithm for the calculation of a constant q transform. *The Journal of the Acoustical Society of America*, 92(5):2698–2701, 1992.

[5] Arshia Cont. Realtime multiple pitch observation using sparse non-negative constraints. In *Proceedings of the 7th International Conference on Music Information Retrieval*, Victoria (BC), Canada, October 8-12 2006. `http://ismir2006.ismir.net/PAPERS/ISMIR06170_Paper.pdf`.

[6] Valentin Emiya, Roland Badeau, and Bertrand David. Multipitch Estimation of Piano Sounds Using a New Probabilistic Spectral Smoothness Principle. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1643–1654, August 2010.

[7] Valentin Emiya, Nancy Bertin, Bertrand David, and Roland Badeau. Maps-a piano database for multipitch estimation and automatic transcription of music. 2010.

[8] Bernard Gold. Computer program for pitch extraction. *The Journal of the Acoustical Society of America*, 34(7):916–921, 1962.

[9] Gene H Golub and Victor Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on numerical analysis*, 10(2):413–432, 1973.

[10] Michael Good. Musicxml for notation and analysis. *The virtual score: representation, retrieval, restoration*, 12:113–124, 2001.

[11] Anssi Klapuri and Manuel Davy, editors. *Signal Processing Methods for Music Transcription*. Springer, New York, 2006.

[12] A.P. Klapuri. A perceptually motivated multiple-F0 estimation method. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005.*, pages 291–294. IEEE, 2005.

[13] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[14] Chuan-bi Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[15] Gareth Loy. Musicians make a standard: the midi phenomenon. *Computer Music Journal*, pages 8–26, 1985.

[16] Matija Marolt. Sonic: Transcription of polyphonic piano music with neural networks. In *Workshop on Current Research Directions in Computer Music*, pages 217–224, 2001.

[17] Matija Marolt. A connectionist approach to automatic transcription of polyphonic piano music. *Multimedia, IEEE Transactions on*, 6(3):439–449, 2004.

[18] James A. Moorer. On the Transcription of Musical Sound by Computer. *Computer music journal*, 1(4):32–38, 1977.

[19] Krištof Oštir. *Daljinsko zaznavanje*. Založba ZRC, 2006.

[20] Matevž Pesek, Aleš Leonardis, and Matija Marolt. A compositional hierarchical model for music information retrieval. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 131–136, Taipei, 2014.

[21] Graham E Poliner and Daniel PW Ellis. A discriminative model for polyphonic piano transcription. *EURASIP Journal on Applied Signal Processing*, 2007(1):154–154, 2007.

[22] Stanisław A. Raczyński, Nobutaka Ono, and Shigeki Sagayama. Multipitch analysis with harmonic nonnegative matrix approximation. In *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 381–386, Vienna, Austria, September 23-27 2007. `http://ismir2007.ismir.net/proceedings/ISMIR2007_p381_raczynski.pdf`.

[23] P. Smaragdis and J.C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No.03TH8684)*, pages 177–180. IEEE, 2003.

[24] Man Mohan Sondhi. New methods of pitch extraction. *Audio and Electroacoustics, IEEE Transactions on*, 16(2):262–266, 1968.

[25] Emmanuel Vincent, Nancy Bertin, and Roland Badeau. Adaptive harmonic spectral decomposition for multiple pitch estimation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(3):528–537, 2010.

[26] Felix Weninger, Christian Kirst, Bjorn Schuller, and Hans-Joachim Bungartz. A discriminative approach to polyphonic piano note transcription using supervised non-negative matrix factorization. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6–10, Vancouver, 2013.